**Mobile Application and Development**

**Manual**

CS

Semester 6th

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roll Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INSTRUCTOR:**

**MUHAMMAD FAHAD**

**Index**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S #** | **DATE** | **Practical** | **Remarks** | **Signature** |
| 1 |  | Lab 1: Introduction |  |  |
| 2 |  | Lab 2: Installation and Configuration |  |  |
| 3 |  | Lab 3: Basic Designing |  |  |
| 4 |  | Lab 4: Simple Addition |  |  |
| 5 |  | Lab 5: User-input addition, subtraction, multiplication and division |  |  |
| 6 |  | Lab 6: Calculation using Radio Button |  |  |
| 7 |  | Lab 7: Simple Calculator(Windows) |  |  |
| 8 |  | Lab 8: Toast, Intent (Calculator) |  |  |
| 9 |  | Lab 9: Quiz |  |  |
| 10 |  | Lab 10: Quiz app activity |  |  |
| 11 |  | Lab 11:Activities (Multiple activities) |  |  |
| 12 |  | Lab 12: Database (SQL Lite) Introduction |  |  |
| 13 |  | Lab 13:Database (SQL Lite) Features |  |  |
| 14 |  | Final Evaluation |  |  |

**Marks Evaluation**

|  |  |
| --- | --- |
| **Experiment No.** | **Marks** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| **Final** |  |

**Instructor Signature**

**LAB # 01**

**INTRODUCTION TO ANDROID STUDIO**

**PURPOSE:**

You learned about characteristics that define Android and Android apps. You looked at user interface components, device buttons, and application components, such as activities and intents

**OBJECTIVES:**

Where the expected achievements by the end of the experiment are stated

**EXERCISES:**

Extra questions and tasks for the student to carry after the lab, and include in

The lab report.

**Meet Android Studio:**

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on [IntelliJ IDEA](https://www.jetbrains.com/idea/). On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

* A flexible Gradle-based build system
* A fast and feature-rich emulator
* A unified environment where you can develop for all Android devices
* Instant Run to push changes to your running app without building a new APK
* Code templates and GitHub integration to help you build common app features and import sample code
* Extensive testing tools and frameworks
* Lint tools to catch performance, usability, version compatibility, and other problems
* C++ and NDK support

## What is Android Studio?

It is an Android focused IDE, designed especially for the Android development. It was launched on 16 May 2013, during Google I/O 2013 annual event. Android studio contains all the Android SDK tools to design, test, debug and profile your app. By looking at the development tools and environment, we can its similar to eclipse with the ADT plug-in but as I have mentioned above its android focused IDE, there are many cool features available in Android Studio, which can foster and increase your development productivity.

One great thing is that it depends on the IntelliJ Idea IDE, which is proved itself a great IDE and has been using by most all the Android engineers.



## Features of Android Studio:

Here are the cool features:

* Powerful code editing (smart editing, code re-factoring)
* Rich layout Editor (As soon as you drag and drop views on the layout, it shows you a preview in all the screens including Nexus 4, Nexus 7, Nexus 10 and many other resolutions. Layout designing can be done much faster way as compared to eclipse.)
* Gradle-based build support
* Maven Support
* Template-based wizards
* [Lint tool](http://developer.android.com/tools/help/lint.html) analysis (The Android lint tool is a static code analysis tool that checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization).

**Project Structure:**

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

* Android app modules
* Library modules
* Google App Engine modules

By default, Android Studio displays your project files in the Android project. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:

* **Manifests**: Contains the AndroidManifest.xml file.
* **Java**: Contains the Java source code files, including JUnit test code.
* **Res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown (in figure 1, it is showing as **Android**).

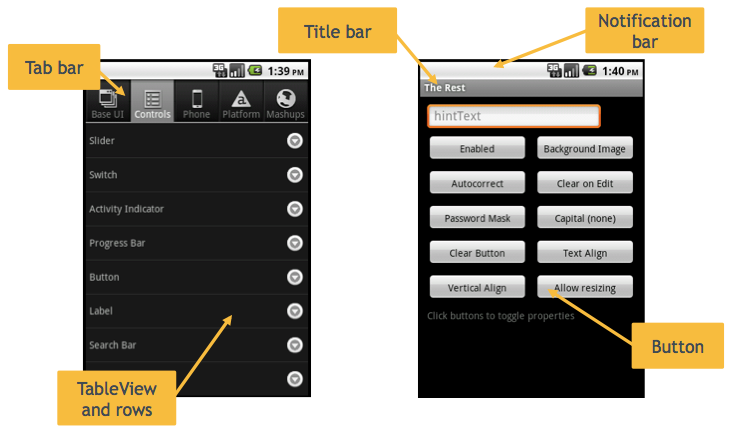
You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the **Problems** view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.

**Android Platform Overview:**

User interface conventions:

You will find quickly when researching Android that the UI can vary significantly among devices. While there is a standard, "vanilla" UI common to all Android operating systems, this is rarely seen on device. This is because Android is an open source, and thus extensible, mobile operating system. Mobile device vendors, like Motorola and HTC, are free to take the base UI and enhance it as they see fit. Android seeks to enable each vendor’s own vision of what Android should be on his or her device, not dictate it.

The following screens illustrate some of the common user interface components offered by the "vanilla" Android UI.

****

#### Hardware buttons:

Android devices feature four dedicated-function "hardware" buttons: Back, Menu, Home, and Search. Depending on the device, these buttons can be physical buttons or touch-based user interface buttons. The location and order of those buttons varies between device vendors.

* **Back** – tap to return to the previous activity in the stack; if none remains in the stack you are returned to the home screen.
* **Home** – return immediately to the home screen, pausing any currently opened apps
* **Menu** – display a menu of activity-specific options
* **Search** – display search functionality, either in-app or system-wide

The Home button behavior cannot be overridden, but you can interact with the Back, Search and Menu buttons.

To override the default behavior for the Back button, add an event listener for the Window. Android back event (Prior to Release 3.0, this event was named Window. android back. The older name is now deprecated.)

To receive an event when the Search button is pressed, add an event listener for the Window. Android search event (Prior to Release 3.0, this event was named Window. Android search the older name is now deprecated.)

You cannot directly override the Menu button, but you can customize the menu displayed when the user presses the Menu button. See Android Menus in the Android UI Components and Conventions section for more information.

#### **Screen sizes and densities:**

#### Android devices vary greatly in screen size and density. **Screen size** represents the physical size of the display. Measured diagonally, it can range from quite small (2.8 inches/71 mm) to large (4.3 inches/110mm) to tablet sizes (10.1 inches/256 mm). Android divides these into generally four categories: small, normal, large, and large, which are then denoted by their density-independent pixel measurements, which Google labels "dp." Each density-independent pixel is equivalent to one physical pixel on a 160 dpi screen.

* small screens are at least 426dp x 320dp
* normal screens are at least 470dp x 320dp
* large screens are at least 640dp x 480dp
* xlarge screens are at least 960dp x 720dp

Aspect ratios vary as well, though Android generally lumps them into two buckets: long and "not long" with the latter corresponding to devices with an aspect ratio not significantly different from the 320 x 470 "normal" screen.

Finally, density describes the actual pixels (aka dots) per square inch resolution of the screen. These range between:

* ldpi screens are roughly 120 dpi
* mdpi screens are roughly 160 dpi (this is the baseline "normal" density)
* tvdpi screens are roughly 213 dpi
* hdpi screens are roughly 240 dpi
* xhdpi screens are roughly 320 dpi

Titanium enables you to simply scale your user interface to fit the device's screen. However, it also offers convenient features for specifically handling assets and layout for various screen sizes. You should plan to test on multiple devices if you want your user interface to be "pixel perfect" on all devices.

There is also a nodpi option where your images will not be scaled by the system if you do not want to create various assets for each density.

### Application Components:

Android applications are built from the following components. Titanium shields you from some of the particulars, though it also gives you the tools to interact with these components when you want to.

* Activities
* Services
* Intents

(We are simplifying things a bit here by ignoring content providers and broadcast receivers. Read Google's Android Fundamentals guide for information that is more detailed.)

#### **Activities:**

An Android app is made up of one or more activities. Each activity represents a "single screen with a user interface." For example, a window that lists messages in an inbox would be an activity. The window in which you read one of those messages would be a separate activity. The set of activities in an app work together to provide the functionality of that app

One of the most powerful features of Android activities is that apps can start each other's activities. Let us say you want the user to be able to snap a photo within your app. You could write all the code to display the camera's live view along with the buttons that make up the photo-snapping experience. However, with Android, you do not have to. The built-in Camera app has an activity that does all that already. All your app needs to do is launch the Camera app's activity and define what should happen with the data that is returned. Other apps can call on the activities that are defined within your app as well.

This shared-activity scheme is a key strength of the Android platform. Apps can share functionality, and they don't even need to know how those other apps work. Your app does not need to know how Camera's activity grabs the photo. You can just deal with the image that is returned. This activity sharing mechanism is what enables the "Share" button functionality included in many Android apps. This is discussed in more detail in the Intent section.

Each activity is listed in the Android Manifest.xml file. Notations in that file describe which activities are published (and thus available for other apps to call on). Titanium lets you create activities – a "heavy weight" window that corresponds to an Android activity. When Titanium’s compile scripts parse the tiapp.xml file and your code, appropriate entries are created in the AndroidManifest.xml file.

An Android Activity is not created until the "heavy weight" window is opened. Before the window is opened, the activity property refers to a plain JS object, which can be used to setup To Android Activity properties. Once the window is opened, the Android Activity is created, and then the activity property can use the To Android. Activity methods.

You will find more info on the Android developer's Activity guide.

#### **Services:**

Services are "long running" app components that run without user interaction. You might use a service to periodically check a network resource or you play music while your app is in the background. Services are not separate threads or processes. They are not a way to offload work from your main application. You can create services by calling on Titanium's To Android Service module.

More information can be found at the Android developer's services page or the To Android Service page.

#### **Intents:**

Intents are messaging objects that hold data passed between activities, sent to or from a background service, or sent by system broadcasts. Intents enable your app to interact with the activities available on the user's device without knowing which apps the user has installed.

Earlier we stated that you could launch another app's activities. In truth, for security reasons your app cannot directly start another app's activities. Instead, your app sends an Intent, which contains a URI to the content and instructions as to how it should be handled. You can create an explicit intent, in which you request that a specific activity be launched. If it is available, Android launches it for on behalf of your app. The more powerful option is to use an implicit intent, which will return a list of all apps available on a mobile device that are capable of handling your Intent.

Think back to the "Share" button functionality described earlier. Your app might publish some text via an intent, thereby sending a request for a list of all the apps that could handle that data. The OS would present a list of suitable apps to the user, who would choose which app to use. The user could select a Twitter client, email app, or any other app that can handle text. With properly formatted Intents, you can add large amounts of functionality to your apps simply by leveraging apps already installed on the device.

**LAB # 02**

**Installation of Android Studio**

**PURPOSE:**

Installation of Android studio.

**OBJECTIVES:**

To learn how to install android studio.

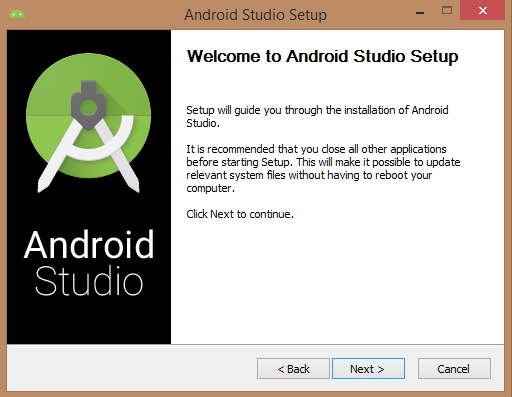
**EXERCISES:**

Extra questions and tasks for the student to carry after the lab, and include in

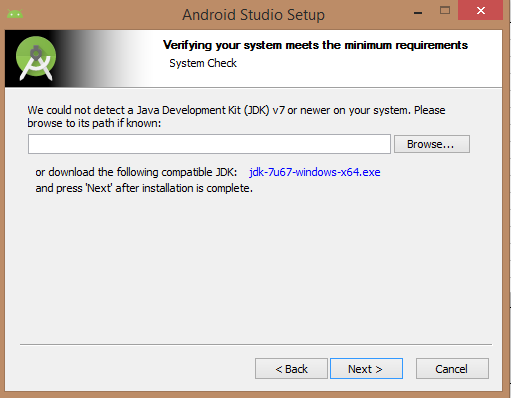
The lab report.

## INSTALLATION:

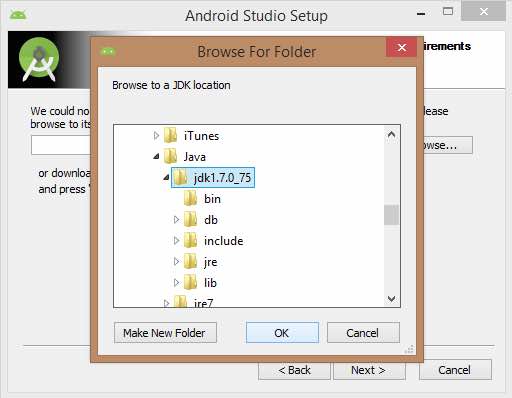
So let us launch *Android Studio.exe* Make sure before launch Android Studio, Our Machine should require installed Java JDK. To install Java JDK, take references of Android environment setup



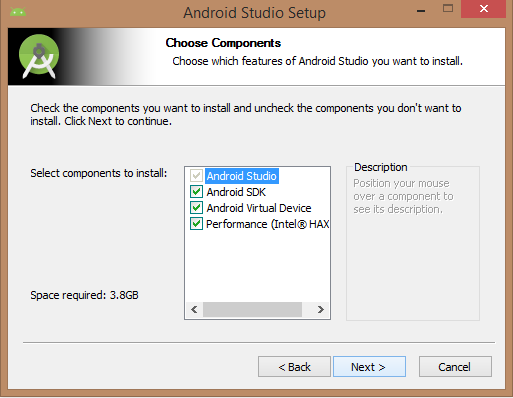
Once you launched Android Studio, it is time to mention JDK7 path or later version in android studio installer.



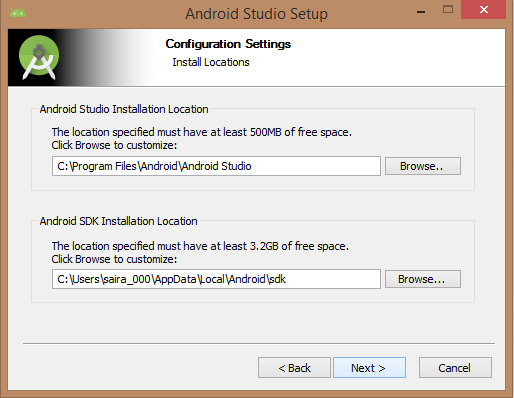
Below the image initiating JDK to android SDK.



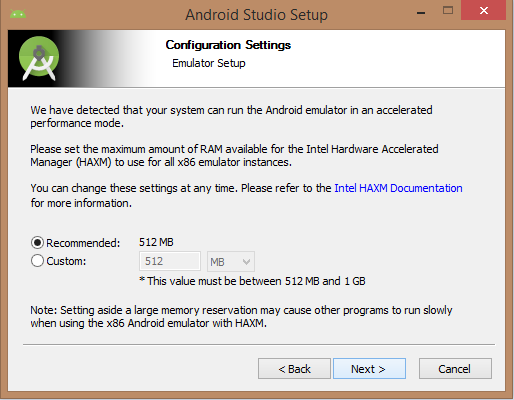
Need to check the components, which are required to create applications, below the image has selected Android Studio, Android SDK, Android Virtual Machine and performance (Intel chip).



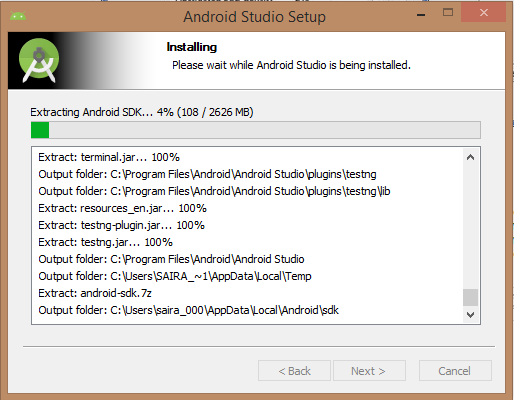
Need to specify the location of local machine path for Android studio and Android SDK, below the image has taken default location of windows 8.1 x64 bit architecture.



Need to specify the ram space for Android emulator by default it would take 512MB of local machine RAM.



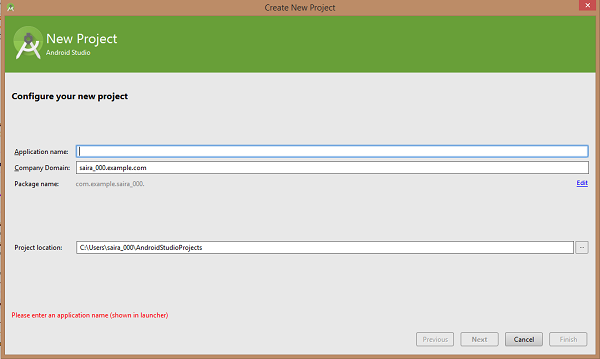
At final stage, it would extract SDK packages into our local machine, it would take a while time to finish the task and would take 2626MB of Hard disk space.



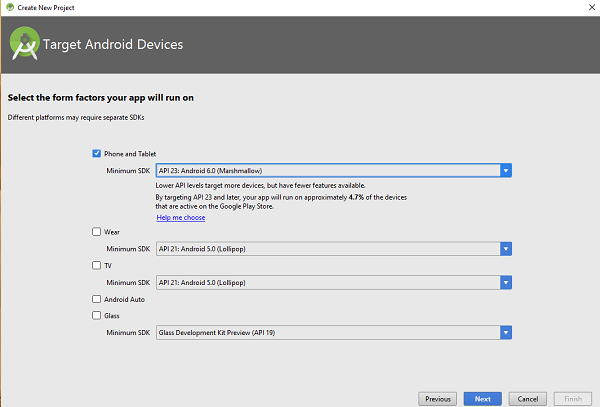
After done all above steps perfectly, you must get finish button and it going to be open android studio project with Welcome to android studio message as shown below



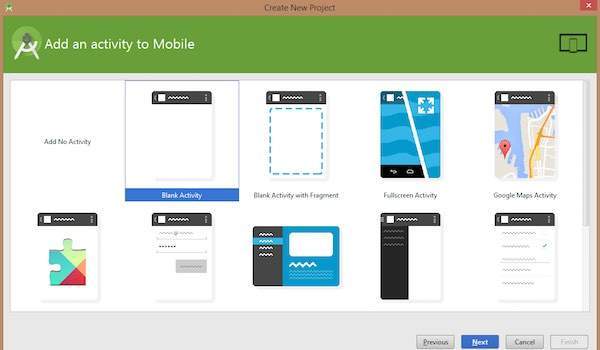
You can start your application development by calling start a new android studio project. In a new installation frame should ask Application name, package information and location of the project.



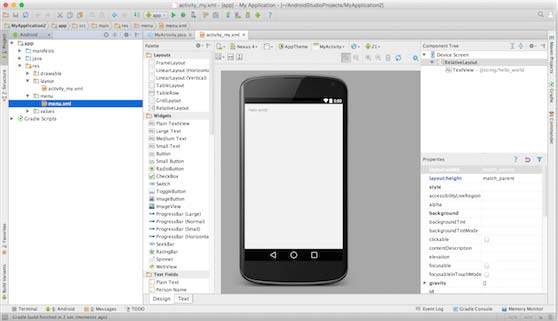
After entered application name, it going to be called select the form factors your application runs on, here need to specify Minimum SDK, in our tutorial, I have declared as API23: Android 6.0(Marshmallow).



The next level of installation should contain selecting the activity to mobile; it specifies the default layout for Applications.

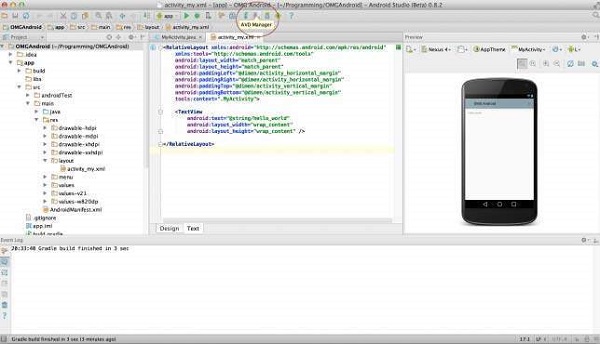


At the final stage, it going to be open development tool to write the application code.

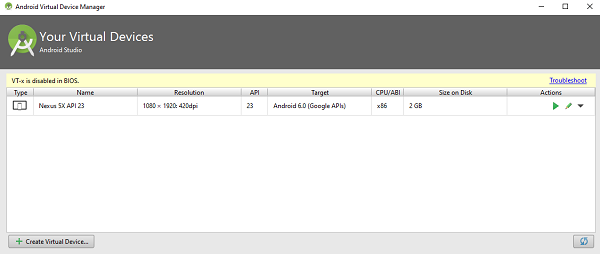


## Create Android Virtual Device:

To test your Android applications, you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager Clicking AVD\_Manager icon as shown below.



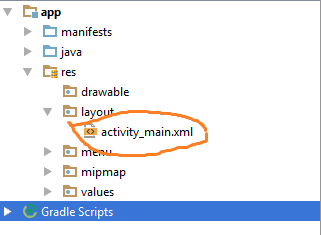
After Click on a virtual device icon, it going to be shown by default virtual devices, which are present on your SDK, or else need to create a virtual device by clicking Create new Virtual device button.



If your AVD is created successfully, it means your environment is ready for Android application development. If you like, you can close this window using top-right cross button. Better, you re-start your machine and once you are done with this last step, you are ready to proceed for your first Android example but before that, we will see few more important concepts related to Android Application Development.

## Hello Word Example:

Before Writing a Hello word code, you must know about XML tags. To write hello word code, you should redirect to App>res>layout>Activity\_main.xml.



To show hello word, we need to call text view with layout (about text view and layout, you must take references at Relative Layout.

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools" android: layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MainActivity">

<TextView android:text="@string/hello\_world"

android:layout\_width="550dp"

android:layout\_height="wrap\_content" />

</RelativeLayout>

Need to run the program by clicking Run>Run App or else need to call shift+f10key. Finally, result should be placed at Virtual devices as shown below.



**LAB # 03**

**BASIC DESIGNNING**

**OBJECTIVES:**

1. To learn basic designing.

2. To learn how to implement it on ANDROID STUDIO.

3. To learn how to use ANDROID STUDIO tools.

**EXERCISES:**

Extra questions and tasks for the student to carry after the lab, and include in

The lab report

**Equipment / requirement:**

* Hardware Requirement:

Personal computer

.

* Software Requirement:

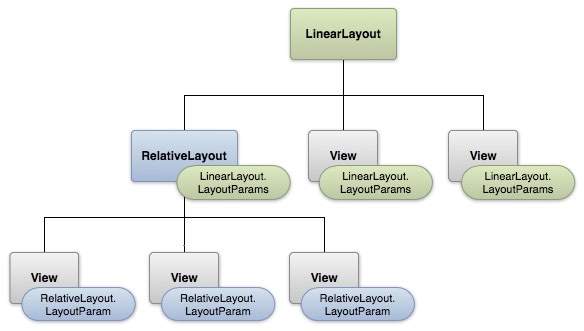
ANDROID STUDIO

**Android - UI Layouts:**

The basic building block for user interface is a **View** object, which is created from the View class, occupies a rectangular area on the screen, and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components like buttons, text fields, etc.

The **View Group** is a subclass of **View** and provides invisible container that hold other Views or other ViewGroups and define their layout properties.

At third level we have different layouts which are subclasses of ViewGroup class and a typical layout defines the visual structure for an Android user interface and can be created either at run time using **View/View Group** objects or you can declare your layout using simple XML file **main\_layout.xml** which is located in the res/layout folder of your project.



## Android Layout Types:

There are number of Layouts provided by Android, which you will use in almost all the Android applications to provide different view, look and feel.

|  |  |
| --- | --- |
| **Sr.No** | **Layout & Description** |
| 1 | [**Linear Layout**](https://www.tutorialspoint.com/android/android_linear_layout.htm)  LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. |
| 2 | [**Relative Layout**](https://www.tutorialspoint.com/android/android_relative_layout.htm)  RelativeLayout is a view group that displays child views in relative positions. |
| 3 | [**Table Layout**](https://www.tutorialspoint.com/android/android_table_layout.htm)  TableLayout is a view that groups views into rows and columns. |
| 4 | [**Absolute Layout**](https://www.tutorialspoint.com/android/android_absolute_layout.htm)  AbsoluteLayout enables you to specify the exact location of its children. |
| 5 | [**Frame Layout**](https://www.tutorialspoint.com/android/android_frame_layout.htm)  The FrameLayout is a placeholder on screen that you can use to display a single view. |
| 6 | [**List View**](https://www.tutorialspoint.com/android/android_list_view.htm)  ListView is a view group that displays a list of scrollable items. |
| 7 | [**Grid View**](https://www.tutorialspoint.com/android/android_grid_view.htm)  GridView is a ViewGroup that displays items in a two-dimensional, scrollable grid. |

## Layout Attributes:

Each layout has a set of attributes, which define the visual properties of that layout. There are few common attributes among all the layouts and there are other attributes, which are specific to that layout. Following are common attributes and will be applied to all the layouts:

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **android:id**  This is the ID which uniquely identifies the view. |
| 2 | **android:layout\_width**  This is the width of the layout. |
| 3 | **android:layout\_height**  This is the height of the layout |
| 4 | **android:layout\_marginTop**  This is the extra space on the top side of the layout. |
| 5 | **android:layout\_marginBottom**  This is the extra space on the bottom side of the layout. |
| 6 | **android:layout\_marginLeft**  This is the extra space on the left side of the layout. |
| 7 | **android:layout\_marginRight**  This is the extra space on the right side of the layout. |
| 8 | **android:layout\_gravity**  This specifies how child Views are positioned. |
| 9 | **android:layout\_weight**  This specifies how much of the extra space in the layout should be allocated to the View. |
| 10 | **android:layout\_x**  This specifies the x-coordinate of the layout. |
| 11 | **android:layout\_y**  This specifies the y-coordinate of the layout. |
| 12 | **android:layout\_width**  This is the width of the layout. |
| 13 | **android:layout\_width**  This is the width of the layout. |
| 14 | **android:paddingLeft**  This is the left padding filled for the layout. |
| 15 | **android:paddingRight**  This is the right padding filled for the layout. |
| 16 | **android:paddingTop**  This is the top padding filled for the layout. |
| 17 | **android:paddingBottom**  This is the bottom padding filled for the layout. |

# **Build a Simple User Interface:**

## Open the Layout Editor:

To get started, set up your workspace as follows:

1. In Android Studio's Project window, open **app > res > layout > activity\_main.xml**.
2. To make more room for the Layout Editor, hide the **Project** window by selecting **View > Tool Windows > Project** (or click **Project** https://developer.android.com/studio/images/buttons/window-project.png on the left side of Android Studio).
3. If your editor shows the XML source, click the **Design** tab at the bottom of the window.
4. Click **Show Blueprint** https://developer.android.com/studio/images/buttons/layout-editor-blueprint.png so only the blueprint layout is visible.
5. Make sure Show Constraints is on. The tooltip in the toolbar should read **Hide Constraints** https://developer.android.com/studio/images/buttons/layout-editor-hide-constraints.png (because they're now showing).
6. Make sure Autoconnect is off. The tooltip in the toolbar should read **Turn On**
7. **Autoconnect** https://developer.android.com/studio/images/buttons/layout-editor-autoconnect-on.png (because it is now off).
8. Click **Default Margins** https://developer.android.com/studio/images/buttons/layout-editor-margin.png in the toolbar and select **16** (you can still adjust the margin for each view later).
9. Click **Device in Editor** https://developer.android.com/studio/images/buttons/layout-editor-device.png in the toolbar and select **Pixel XL**.

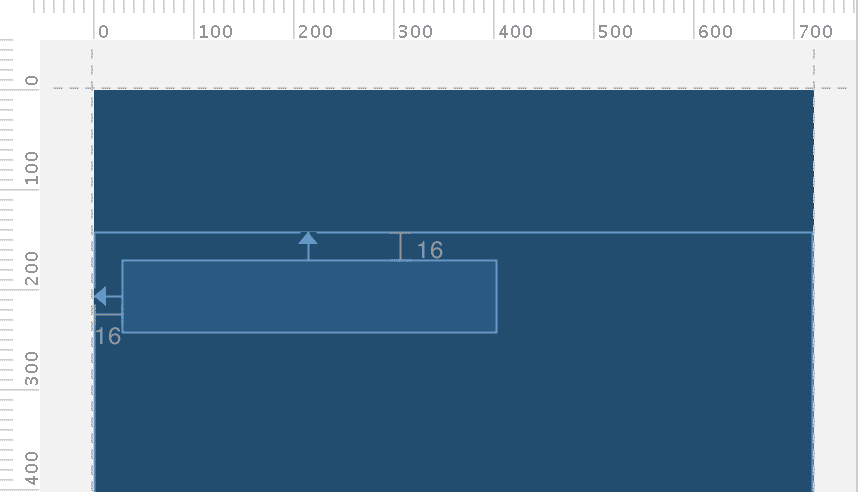
# layout-editor_2x

## Add a text box:

1. First, you need to remove what is already in the layout. So click **TextView** in the **Component Tree** window, and then press Delete.
2. From the **Palette** window on the left, click **Text** in the left pane, and then drag **Plain Text** into the design editor and drop it near the top of the layout. This EditText widget accepts plain text input.
3. Click the view in the design editor. You can now see the resizing handles on each corner (squares), and the constraint anchors on each side (circles).

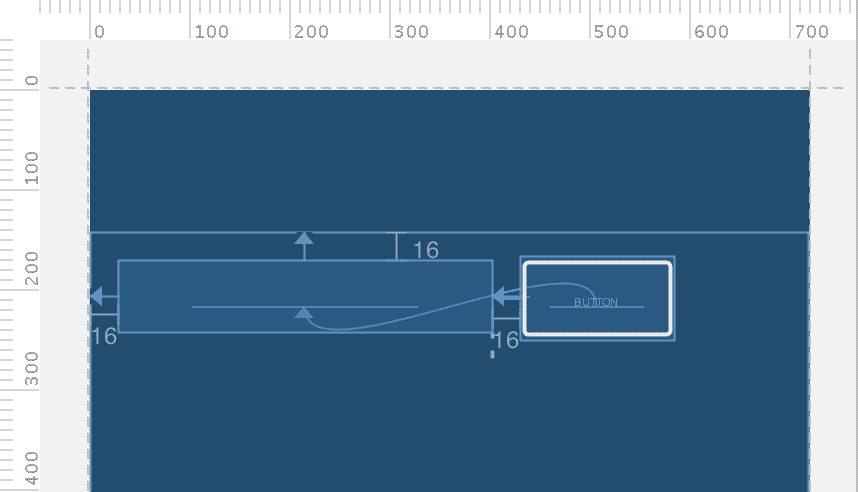
For better control, you might want to zoom in on the editor to 75% or higher using the buttons in the toolbar.

1. Click-and-hold the anchor on the top side, and then drag it up until it snaps to the top of the layout and release. That is a constraint—it specifies the view should be 16dp from the top of the layout (because you set the default margins to 16dp).
2. Similarly, create a constraint from the left side of the view to the left side of the layout.

****

Add a button:

1. From the **Palette**window, click **Widgets** in the left pane, and then drag **Button** into the design editor and drop it near the right side.
2. Create a constraint from the left side of the button to the right side of the text box.
3. To constrain the views in a horizontal alignment, you need to create a constraint between the text baselines. So click the button, and then click **Baseline Constraint**https://developer.android.com/studio/images/buttons/layout-editor-action-baseline.png, which appears in the design editor directly below the selected view. The baseline anchor appears inside the button. Click-and-hold on this anchor and then drag it to the baseline anchor that appears in the text box.



## Change the UI strings:

Click **Show Design** https://developer.android.com/studio/images/buttons/layout-editor-design.png in the toolbar to preview the UI. Notice that the text input is pre-filled with "Name" and the button is labeled "Button." So now, you will change these strings.

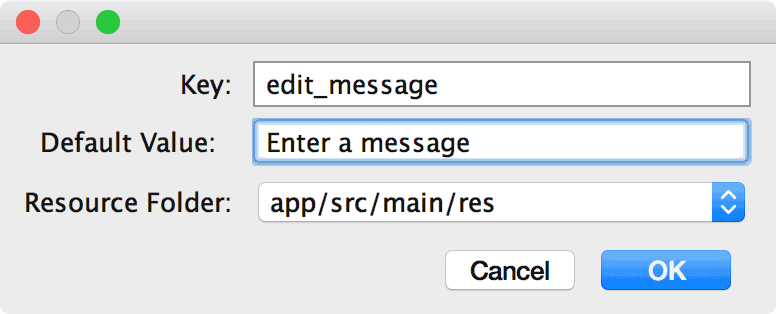
1. Open the **Project**window and then select **res > values > strings.xml**.

This is a string resources file where you should specify all your UI strings. Doing so allows you to manage all UI strings in a single location, which makes it easier to find, update, and localize (compared to hard-coding strings in your layout or app code).

1. Click **Open editor** at the top of the editor window. This opens the Translations Editor, which provides a simple interface for adding and editing your default strings, and helps keep all your translated strings organized.
2. Click **Add Key** https://developer.android.com/studio/images/buttons/add-sign-green-icon.png to create a new string as the "hint text" for the text box.
   1. Enter "edit\_message" for the key name.
   2. Enter "Enter a message" for the value.
   3. Click **OK**.
3. Add another key named "button\_send" with a value of "Send."

Now you can set these strings for each view. So return to the layout file by clicking**activity\_main.xml**in the tab bar, and add the strings as follows:

1. Click the text box in the layout and, if the **Properties** window is not already visible on the right, click **Properties** https://developer.android.com/studio/images/buttons/window-properties.png on the right sidebar.
2. Locate the **hint**property and then click **Pick a Resource** https://developer.android.com/studio/images/buttons/pick-resource.png to the right of the text box. In the dialog that appears, double-click on **edit\_message** from the list.
3. Still viewing the text box properties, also delete the value for the **text** property (currently set to "Name").
4. Now click the button in the layout, locate the **text** property, click **Pick a Resource**, and then select **button\_send**.



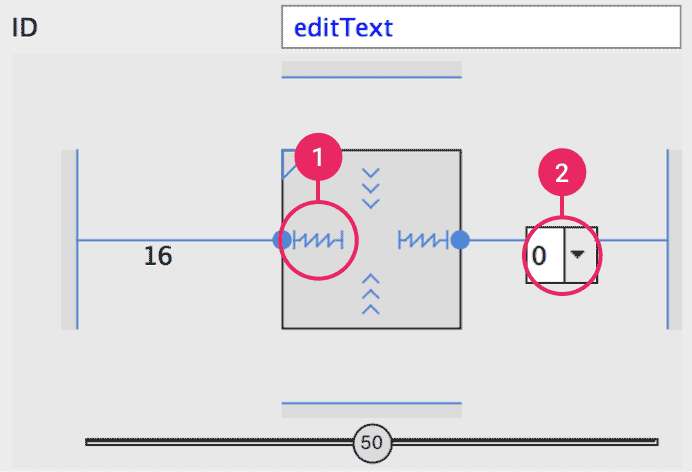
## Make the text box size flexible

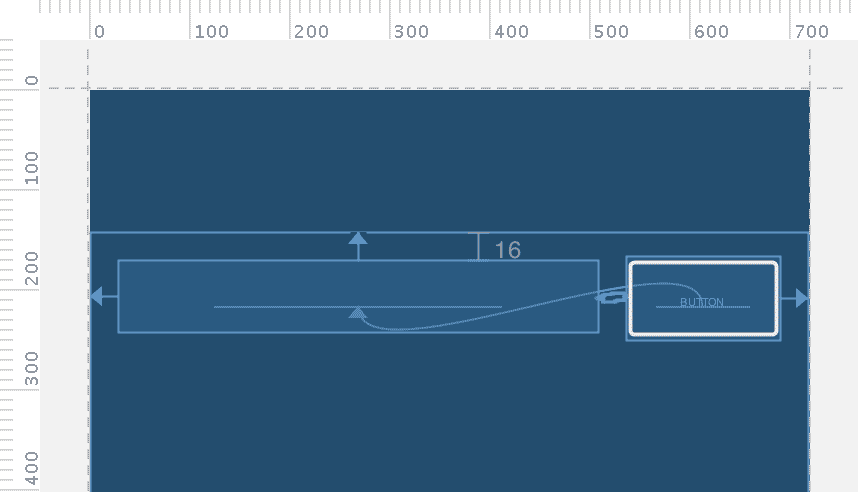
To create a layout that is responsive to different screen sizes, you will now make the text box stretch to fill all remaining horizontal space (after accounting for the button and margins).

1. Create a constraint from the right side of the button to the right side of the parent layout. This now defines the total width that the two views have available (which you can now fill with the text box).
2. Add a constraint from the right side of the text box to the left side of the button. It might look like that is already there, but you are actually adding a bidirectional constraint between the two views. Therefore, both views are constrained to each other. This is called a *chain* (as indicated by the chain between the views), and it enables some extra layout options.
3. Open the **Properties** window for the text box and then click the width indicator until set to **Match Constraints**, as indicated by callout 1 in figure 8. "Match constraints" means that the width is now determined by the horizontal constraints and margins. Therefore, the text box stretches to fill the horizontal space.
4. However, the two views are separated by 32dp instead of 16dp, because both views have margins. So while still viewing the text box properties, change the right margin to 0, as indicated by callout 2 in figure 8.

Now the layout is done and should appear as shown in figure 9.

If it seems your layout did not turn out as expected, click below to see what you’re the XML should look like and compare it to what you see in the **Text** tab.

****

****

## Run the app:

If your app is already installed on the device from [the previous lesson](https://developer.android.com/training/basics/firstapp/running-app.html), simply click **Apply Changes** https://developer.android.com/studio/images/buttons/toolbar-apply-changes.png in the toolbar to update the app with the new layout. Alternatively, click **Run** https://developer.android.com/studio/images/buttons/toolbar-run.png to install and run the app.

**LAB # 04**

**SIMPLE ADDITION OF TWO NUMBERS:**

**PURPOSE:**

Addition of two numbers, and some basic tools use.

**OBJECTIVES:**

To learn how to add two numbers on android studio. In addition, how known of android tools.

**EXERCISES:**

Extra questions and tasks for the student to carry after the lab, and include in

The lab report.

**Android Development Tools List:**

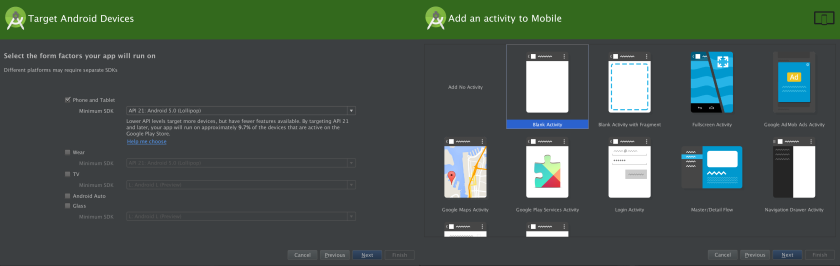
Starting out in Android development can be a daunting task – there is so much information out there, so many tutorials and so many resources it can be hard to navigate.

That is why we have compiled this big list of Android development resources and tools so you can find all you might need in one place.

From IDEs to learning tutorials, consoles to libraries, we have covered the very first steps a budding developer can take on this path, to useful libraries and plug-ins for the more experienced users. The sections in this guide are:

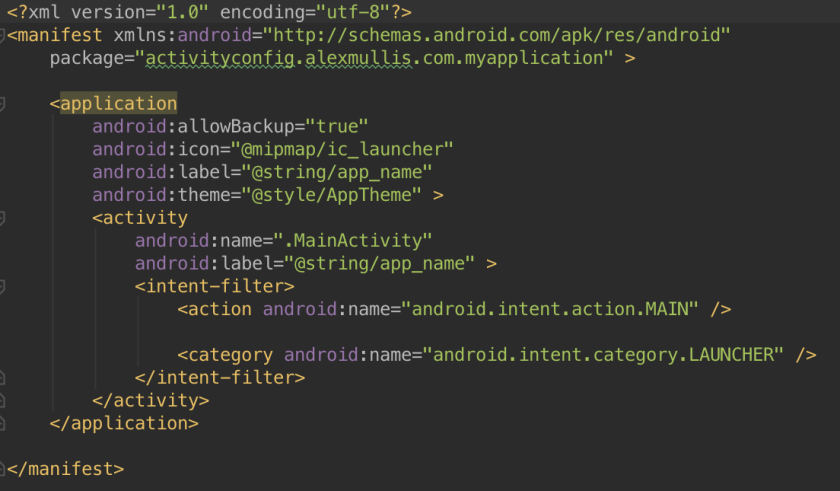
* [Editors and IDEs](http://www.businessofapps.com/guide/android-development-tools/#1)
* [Language Resources](http://www.businessofapps.com/guide/android-development-tools/#2)
* [Libraries](http://www.businessofapps.com/guide/android-development-tools/#3)
* Plugins

## The Basic:



Android Studio replaced Eclipse as the main IDE for Android development in 2014. With this change, Google has revamped the way developers can take advantage of all the Android development tools.

One of these improvements is the way setting up a new project works. It has never been easier to set up a new project, just click File>New>New Project and Android Studio will step you through exactly what you need to get started. Once this step is completed, click on File>New Module to create the actual Graphical User Interface of the app. This creates a new Module with a few folders and other files. Starting with the “manifests” folder, which holds the AndroidManifest.xml. The file holds basic information including the name of the app that shows up on the device and any permission that you define. Here is a sample AndroidManifest.xml:



**CREATE NEW ANDROID PROJECT:**

**XML:**  
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout\_width="fill\_parent"  
    android:layout\_height="fill\_parent"  
    android:orientationss="vertical" >  
  
    <TextView  
        android:layout\_width="fill\_parent"  
        android:layout\_height="wrap\_content"  
        android:gravity="center"  
        android:text="Maths Function" />  
    <EditText  
        android:id="@+id/editText1"  
        android:layout\_width="match\_parent"  
        android:layout\_height="wrap\_content"  
        android:inputType="number" >  
        <requestFocus />  
    </EditText>  
    <EditText  
        android:id="@+id/editText2"  
        android:layout\_width="match\_parent"  
        android:layout\_height="wrap\_content"  
        android:inputType="number" />  
    <Button  
        android:id="@+id/buttonsum"  
        android:layout\_width="match\_parent"  
        android:layout\_height="wrap\_content"  
        android:text="Sum/Addition" />

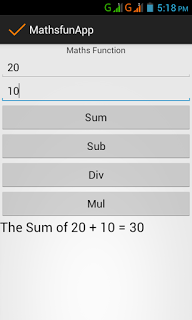
     <Button  
         android:id="@+id/buttonsub"  
         android:layout\_width="match\_parent"  
         android:layout\_height="wrap\_content"  
         android:text="Subtraction" />  
     <Button  
         android:id="@+id/buttondiv"  
         android:layout\_width="match\_parent"  
         android:layout\_height="wrap\_content"  
         android:text="Division" />  
     <Button  
         android:id="@+id/buttonmul"  
         android:layout\_width="match\_parent"  
         android:layout\_height="wrap\_content"  
         android:text="multiplication" />  
    <TextView  
        android:id="@+id/textView1"  
        android:layout\_width="match\_parent"  
        android:layout\_height="wrap\_content"  
        android:text="Large Text"  
        android:textAppearance="?android:attr/textAppearanceLarge" />  
</LinearLayout>

**Main Activity:**  
package dev.androidapplink.mathsfunapp;  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;

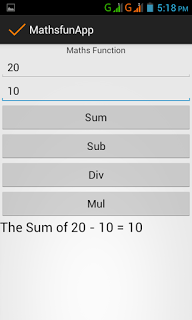
public class MainActivity extends Activity {  
 /\*\* Called when the activity is first created. \*/  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
//Create object  
 Button btnsum = (Button) findViewById(R.id.buttonsum);  
 Button btnsub = (Button) findViewById(R.id.buttonsub);  
 Button btndiv = (Button) findViewById(R.id.buttondiv);  
 Button btnmul = (Button) findViewById(R.id.buttonmul);   
 final EditText etv = (EditText) findViewById(R.id.editText1);  
 final EditText etv2 = (EditText) findViewById(R.id.editText2);  
 final TextView result = (TextView) findViewById(R.id.textView1);  
 // Create button click event  
 btnsum.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 int x = new Integer(etv.getText().toString());  
 int y = new Integer(etv2.getText().toString());  
 int sum = x + y; //Perform Maths operation  
 result.setText("The ANS of " + x + " + " + y + " = " + sum);//print answer  
 }  
 });  
 btnsub.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 int x = new Integer(etv.getText().toString());  
 int y = new Integer(etv2.getText().toString());  
 int sub = x - y; //Perform Maths operation  
 result.setText("The ANS of " + x + " - " + y + " = " + sub);//print answer  
 }  
 });  
 btndiv.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 int x = new Integer(etv.getText().toString());  
 int y = new Integer(etv2.getText().toString());  
 int div = x / y; //Perform Maths operation  
 result.setText("The ANS of " + x + " / " + y + " = " + div);//print answer  
 }  
 });  
 btnmul.setOnClickListener(new OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 int x = new Integer(etv.getText().toString());  
 int y = new Integer(etv2.getText().toString());  
 int mul = x \* y; //Perform Maths operation  
 result.setText("The ANS of " + x + " \* " + y + " = " + mul);//Print answer  
 }  
 });  
 }}

**OUTPUT:**

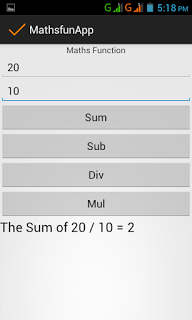
**Addition/Sum:**

****

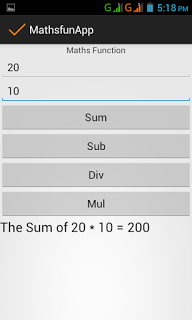
**Subtraction:**

****

**Division:**

****

**Multiplication:**

****

**LAB # 05**

**ADDITION, SUBTRACTION, MULTIPLICATION AND DIVISION (USER-INPUT)**

**CODING:**

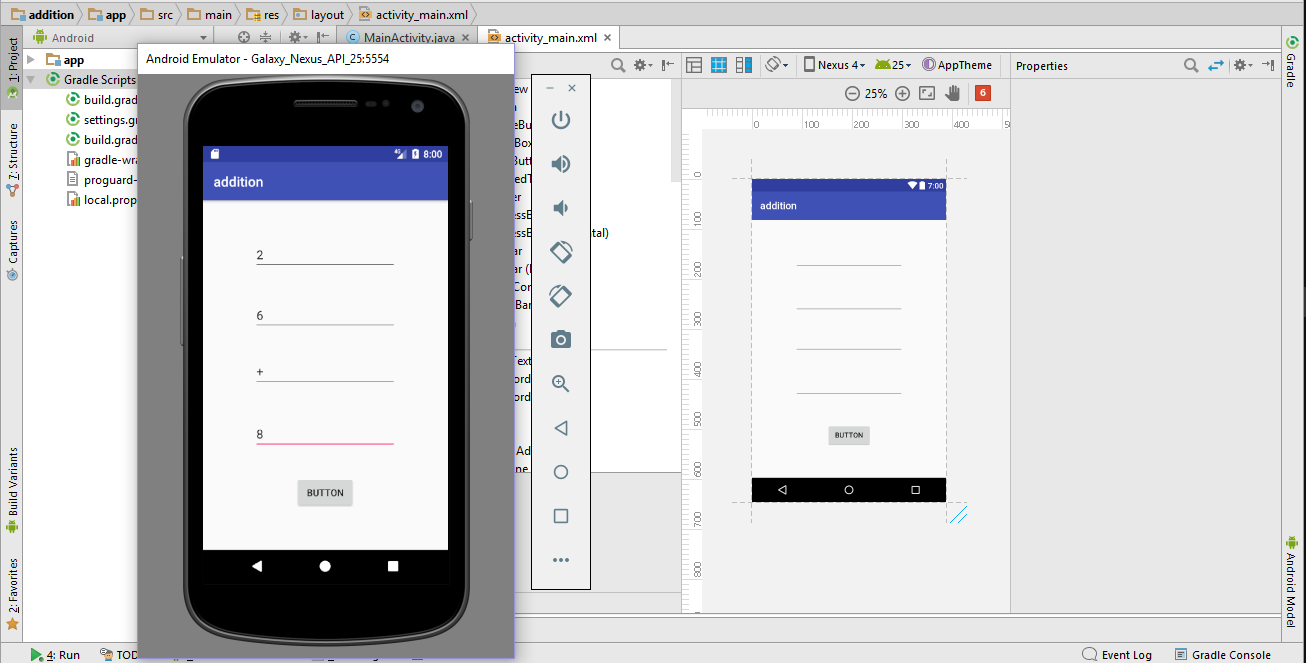
**MAIN ACTIVITY:**

**package** com.example.newapp.addition;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.EditText;  
**public class** MainActivity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
 **public void** calculation(View v){  
 EditText txt1=(EditText)findViewById(R.id.***editText***);  
 EditText txt2=(EditText)findViewById(R.id.***editText2***);  
 EditText txtres=(EditText)findViewById(R.id.***editText4***);  
 EditText opt=(EditText)findViewById(R.id.***editText3***);  
 **int** fno,sno,res=0;  
 String operator;  
 fno=Integer.*parseInt*(txt1.getText().toString());  
 sno=Integer.*parseInt*(txt2.getText().toString());  
 *// int result=Integer.parseInt(txtres.getText().toString());* operator=opt.getText().toString();  
 *//txtres.setText(Integer.toString(result));* **if** (operator.equals(**"+"**)){  
 res=fno+sno;  
 }  
 **else if** (operator.equals(**"-"**)){  
 res=fno-sno;  
 }  
 **else if**(operator.equals(**"\*"**)){  
 res=fno\*sno;  
 }  
 **else if**(operator.equals(**"/"**)){  
 res=fno/sno;  
 }  
 txtres.setText(Integer.*toString*(res));  
 }  
}

**XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.newapp.addition.MainActivity"**>  
 <**RelativeLayout  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**>  
 <**EditText  
 android:id="@+id/editText2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignStart="@+id/editText"  
 android:layout\_below="@+id/editText"  
 android:layout\_marginTop="43dp"  
 android:ems="10"  
 android:inputType="textPersonName"** />  
  
 <**EditText  
 android:id="@+id/editText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/editText12"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="57dp"  
 android:ems="10"  
 android:inputType="textPersonName"** />  
 <**EditText  
 android:id="@+id/editText3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="37dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:layout\_below="@+id/editText2"  
 android:layout\_alignStart="@+id/editText2"** />  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginBottom="60dp"  
 android:onClick="calculation"  
 android:text="Button"** />  
 <**EditText  
 android:id="@+id/editText4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignStart="@+id/editText3"  
 android:layout\_below="@+id/editText3"  
 android:layout\_marginTop="46dp"  
 android:ems="10"  
 android:inputType="textPersonName"** />  
 </**RelativeLayout**>  
</**android.support.constraint.ConstraintLayout**>

**Output:**

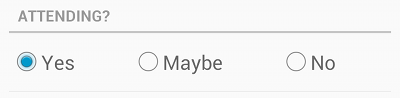


**LAB # 06**

**CALCULATION USING RADIO BUTTON**

**RADIO BUTTON:**

Radio buttons allow the user to select one option from a set. You should use radio buttons for optional sets that are mutually exclusive if you think that the user needs to see all available options side-by-side. If it is not necessary to show all options side-by-side, use a [spinner](https://developer.android.com/guide/topics/ui/controls/spinner.html) instead.



**CODING:**

**MAIN ACTIVITY:**

package com.as400samplecode;

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.TextView;

public class AndroidRadioButtonsActivity extends Activity {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

RadioButton iOS = (RadioButton) findViewById(R.id.selectIOS);

iOS.setChecked(true);

checkButtonClick();

}

private void checkButtonClick() {

Button myButton = (Button) findViewById(R.id.button1);

myButton.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

//get selected radio button from radioGroup

RadioGroup radioGroup = (RadioGroup) findViewById(R.id.osGroup);

int selectedId = radioGroup.getCheckedRadioButtonId();

//find the radio button by returned id

RadioButton osButton = (RadioButton) findViewById(selectedId);

StringBuffer responseText = new StringBuffer();

responseText.append("The following Radio button is selected...\n");

responseText.append(osButton.getText());

TextView myTextView = (TextView) findViewById(R.id.responseText);

myTextView.setText(responseText);

}

});

}

}

**XML:**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:text="@string/hello"

android:textSize="20sp" />

<RadioGroup

android:id="@+id/osGroup"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" >

<RadioButton

android:id="@+id/selectAndroid"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:checked="true"

android:text="Android" />

<RadioButton

android:id="@+id/selectIOS"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="iOS" />

<RadioButton

android:id="@+id/selectWindows"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Windows" />

<RadioButton

android:id="@+id/selectRIM"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="RIM" />

</RadioGroup>

<Button

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:text="Click here to see Results" />

<TextView

android:id="@+id/responseText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

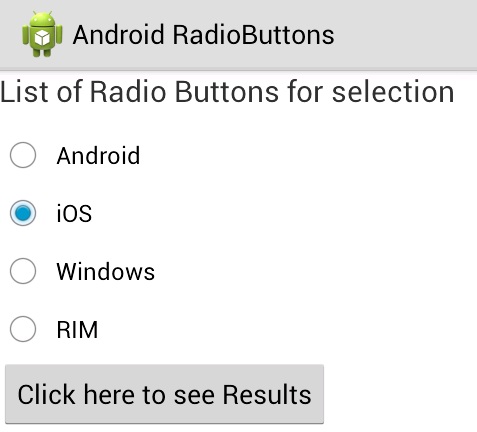
android:padding="5dp"

android:text=""

android:textAppearance="?android:attr/textAppearanceMedium" />

</LinearLayout>

**OUTPUT:**



**LAB # 07**

**SIMPLE CALCULATOR (WINDOWS)**

**CODING:**

**MAIN ACTIVITY:**

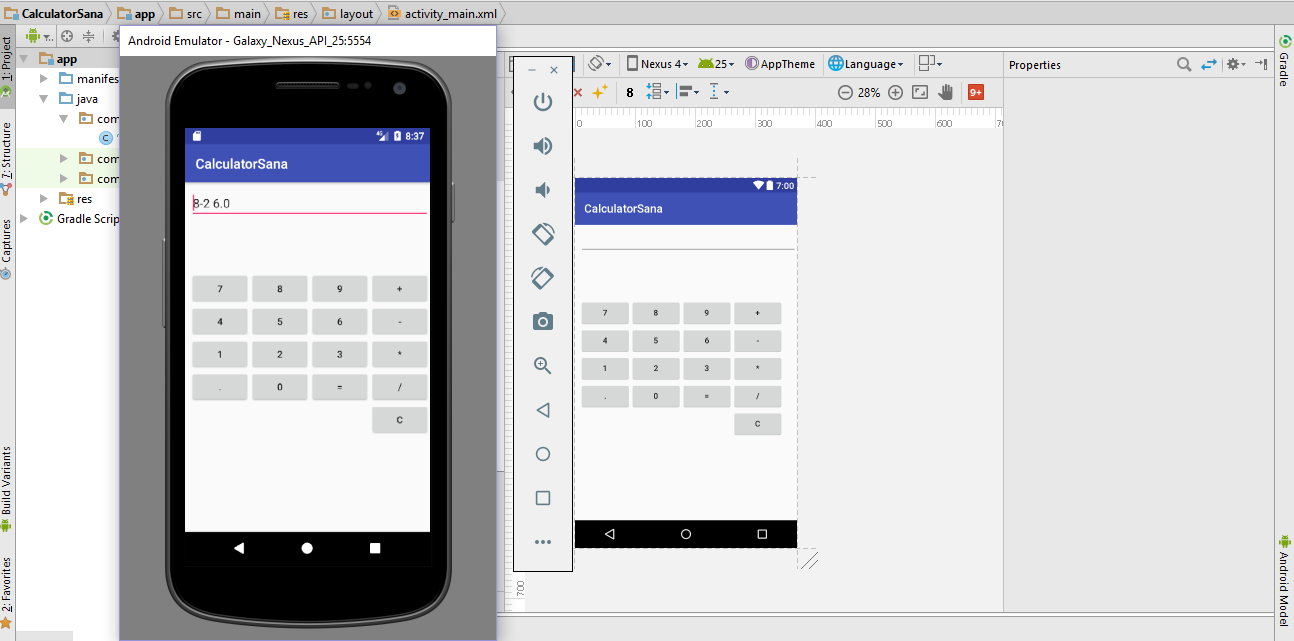
**package** com.example.newapp.calculatorsana;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** java.util.regex.Pattern;  
**public class** MainActivity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **\_screen** = (EditText) findViewById(R.id.***editText***);  
 **\_screen**.setText(**display**);  
 }  
 **private** EditText **\_screen**;  
 **private** String **display** = **""**;  
 **private** String **currentOperator** = **""**;  
 **private** String **result** = **""**;  
 **private void** updateScreen(){  
 **\_screen**.setText(**display**);  
 }  
 **public void** onClickNumber(View v){  
 **if**(**result** != **""**){  
 clear();  
 updateScreen();  
 }  
 Button b = (Button) v;  
 **display** += b.getText();  
 updateScreen();  
 }  
 **private boolean** isOperator(**char** op){  
 **switch** (op){  
 **case '+'**:  
 **case '-'**:  
 **case 'x'**:  
 **case '÷'**:**return true**;  
 **default**: **return false**;  
 }  
 }  
 **public void** onClickOperator(View v){  
 **if**(**display** == **""**) **return**;  
 Button b = (Button)v;  
 **if**(**result** != **""**){  
 String \_display = **result**;  
 clear();  
 **display** = \_display;  
 }  
 **if**(**currentOperator** != **""**){  
 Log.*d*(**"CalcX"**, **""**+**display**.charAt(**display**.length()-1));  
 **if**(isOperator(**display**.charAt(**display**.length()-1))){  
 **display** = **display**.replace(**display**.charAt(**display**.length()-1), b.getText().charAt(0));  
 updateScreen();  
 **return**;  
 }**else**{  
 getResult();  
 **display** = **result**;  
 **result** = **""**;  
 }  
 **currentOperator** = b.getText().toString();  
 }  
 **display** += b.getText();  
 **currentOperator** = b.getText().toString();  
 updateScreen();  
 }  
 **private void** clear(){  
 **display** = **""**;  
 **currentOperator** = **""**;  
 **result** = **""**;  
 }  
 **public void** onClickClear(View v){  
 clear();  
 updateScreen();  
 }  
 **private double** operate(String a, String b, String op){  
 **switch** (op){  
 **case "+"**: **return** Double.*valueOf*(a) + Double.*valueOf*(b);  
 **case "-"**: **return** Double.*valueOf*(a) - Double.*valueOf*(b);  
 **case "x"**: **return** Double.*valueOf*(a) \* Double.*valueOf*(b);  
 **case "÷"**: **try**{  
 **return** Double.*valueOf*(a) / Double.*valueOf*(b);  
 }**catch** (Exception e){  
 Log.*d*(**"Calc"**, e.getMessage());  
 }  
 **default**: **return** -1;  
 }  
 }  
 **private boolean** getResult(){  
 **if**(**currentOperator** == **""**) **return false**;  
 String[] operation = **display**.split(Pattern.*quote*(**currentOperator**));  
 **if**(operation.**length** < 2) **return false**;  
 **result** = String.*valueOf*(operate(operation[0], operation[1], **currentOperator**));  
 **return true**;  
 }  
 **public void** onClickEqual(View v){  
 **if**(**display** == **""**) **return**;  
 **if**(!getResult()) **return**;  
 **\_screen**.setText(**display** + **'\n'** + String.*valueOf*(**result**));  
 }  
}

**XML:**

|  |
| --- |
|  |

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.newapp.calculatorsana.MainActivity"**>  
 <**RelativeLayout  
 android:layout\_width="0dp"  
 android:layout\_height="495dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginLeft="8dp"**>  
 <**EditText  
 android:id="@+id/editText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:ems="10"  
 android:inputType="textPersonName"** />  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/editText"  
 android:layout\_marginTop="78dp"  
 android:onClick="onClickNumber"  
 android:text="7"** />  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignTop="@+id/button"  
 android:layout\_toEndOf="@+id/button"  
 android:onClick="onClickNumber"  
 android:text="8"** />  
 <**Button  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignTop="@+id/button2"  
 android:layout\_toEndOf="@+id/button2"  
 android:onClick="onClickNumber"  
 android:text="9"** />  
 <**Button  
 android:id="@+id/button4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/button"  
 android:onClick="onClickNumber"  
 android:text="4"** />  
 <**Button  
 android:id="@+id/button5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/button2"  
 android:layout\_toEndOf="@+id/button4"  
 android:onClick="onClickNumber"  
 android:text="5"** />  
 <**Button  
 android:id="@+id/button6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/button3"  
 android:layout\_toEndOf="@+id/button5"  
 android:onClick="onClickNumber"  
 android:text="6"** />  
 <**Button  
 android:id="@+id/button7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignTop="@+id/button3"  
 android:layout\_toEndOf="@+id/button3"  
 android:onClick="onClickOperator"  
 android:text="+"** />  
 <**Button  
 android:id="@+id/button8"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/button7"  
 android:layout\_toEndOf="@+id/button6"  
 android:onClick="onClickOperator"  
 android:text="-"** />  
 <**Button  
 android:id="@+id/button9"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/button4"  
 android:onClick="onClickNumber"  
 android:text="1"** />  
 <**Button  
 android:id="@+id/button10"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/button5"  
 android:layout\_toEndOf="@+id/button9"  
 android:onClick="onClickNumber"  
 android:text="2"** />  
 <**Button  
 android:id="@+id/button11"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/button6"  
 android:layout\_toEndOf="@+id/button10"  
 android:onClick="onClickNumber"  
 android:text="3"** />  
 <**Button  
 android:id="@+id/button12"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/button8"  
 android:layout\_toEndOf="@+id/button11"  
 android:onClick="onClickOperator"  
 android:text="\*"** />  
 <**Button  
 android:id="@+id/button13"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onClickNumber"  
 android:text="."  
 android:layout\_alignBaseline="@+id/button14"  
 android:layout\_alignBottom="@+id/button14"  
 android:layout\_alignParentStart="true"** />  
 <**Button  
 android:id="@+id/button14"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onClickNumber"  
 android:text="0"  
 android:layout\_below="@+id/button10"  
 android:layout\_toEndOf="@+id/button13"** />  
 <**Button  
 android:id="@+id/button18"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onClickEqual"  
 android:text="="  
 android:layout\_alignBaseline="@+id/button14"  
 android:layout\_alignBottom="@+id/button14"  
 android:layout\_toEndOf="@+id/button14"** />  
 <**Button  
 android:id="@+id/button16"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignBaseline="@+id/button18"  
 android:layout\_alignBottom="@+id/button18"  
 android:layout\_toEndOf="@+id/button18"  
 android:onClick="onClickOperator"  
 android:text="/"** />  
 <**Button  
 android:id="@+id/button17"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onClickClear"  
 android:text="C"  
 android:layout\_below="@+id/button14"  
 android:layout\_toEndOf="@+id/button11"** />  
 </**RelativeLayout**>  
</**android.support.constraint.ConstraintLayout**>

**OUTPUT:**

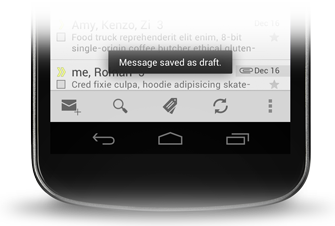


**LAB # 08**

**TOAST, INTENT (CALCULATOR)**

**TOAST:**

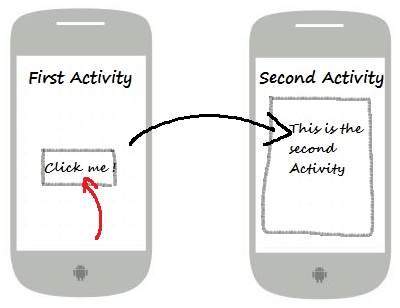
A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. For example, navigating away from an email before you send it triggers a "Draft saved" toast to let you know that you can continue editing later. Toasts automatically disappear after a timeout.

****

**INTENT:**

An intent is an abstract description of an operation to be performed. It can be used with [startActivity](https://developer.android.com/reference/android/content/Context.html" \l "startActivity(android.content.Intent)) to launch an [Activity](https://developer.android.com/reference/android/app/Activity.html), [broadcastIntent](https://developer.android.com/reference/android/content/Context.html" \l "sendBroadcast(android.content.Intent)) to send it to any interested [Broadcast Receiver](https://developer.android.com/reference/android/content/BroadcastReceiver.html) components, and [startService(Intent)](https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)) or [bindService(Intent, ServiceConnection, int)](https://developer.android.com/reference/android/content/Context.html" \l "bindService(android.content.Intent, android.content.ServiceConnection, int)) to communicate with a background [Service](https://developer.android.com/reference/android/app/Service.html).

An Intent provides a facility for performing late runtime binding between the codes in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is a passive data structure holding an abstract description of an action to be performed.

****

**CODING:**

**MAIN ACTIVITY:**

**package** com.example.newapp.task1;  
**import** android.content.Intent;  
**import** android.provider.MediaStore;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.RadioButton;  
**import** android.widget.Toast;  
**public class** MainActivity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
 **public void** calculation(View v) {  
 EditText txt1 = (EditText) findViewById(R.id.***editText***);  
 EditText txt2 = (EditText) findViewById(R.id.***editText2***);  
 *// EditText txtres=(EditText)findViewById(R.id.editText4);* RadioButton r = (RadioButton) findViewById(R.id.***radioButton2***);  
 RadioButton s = (RadioButton) findViewById(R.id.***radioButton***);  
 RadioButton t = (RadioButton) findViewById(R.id.***radioButton3***);  
 RadioButton u = (RadioButton) findViewById(R.id.***radioButton5***);  
 **int** fno, sno, res = 0;  
 String operator;  
 fno = Integer.*parseInt*(txt1.getText().toString());  
 sno = Integer.*parseInt*(txt2.getText().toString());  
 **if** (r.isChecked()) {  
 res = fno + sno;  
 Toast.*makeText*(MainActivity.**this**, Integer.*toString*(res), Toast.***LENGTH\_LONG***).show();  
 } **else if** (s.isChecked()) {  
 res = fno - sno;  
 Toast.*makeText*(MainActivity.**this**, Integer.*toString*(res), Toast.***LENGTH\_LONG***).show();  
 } **else if** (t.isChecked()) {  
 res = fno \* sno;  
 Toast.*makeText*(MainActivity.**this**, Integer.*toString*(res), Toast.***LENGTH\_LONG***).show();  
 } **else if** (u.isChecked()) {  
 res = fno / sno;  
 Toast.*makeText*(MainActivity.**this**, Integer.*toString*(res), Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 **public void** onClick(View v){  
 Intent intent = **new** Intent(MainActivity.**this**, Main2Activity.**class**);  
 startActivity(intent);  
 }  
}

**SECOND MAIN ACTIVITY:**

**package** com.example.newapp.task1;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**public class** Main2Activity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main2***);  
 }  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(Main2Activity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 }  
}

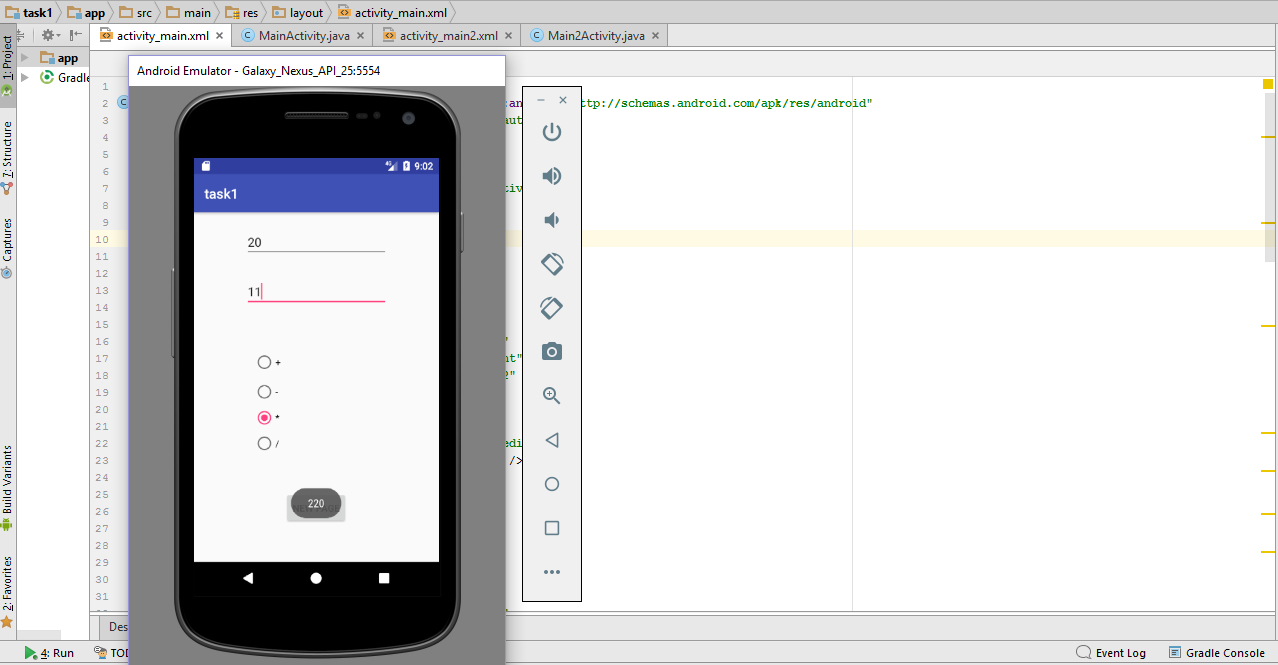
**FIRST XML:**

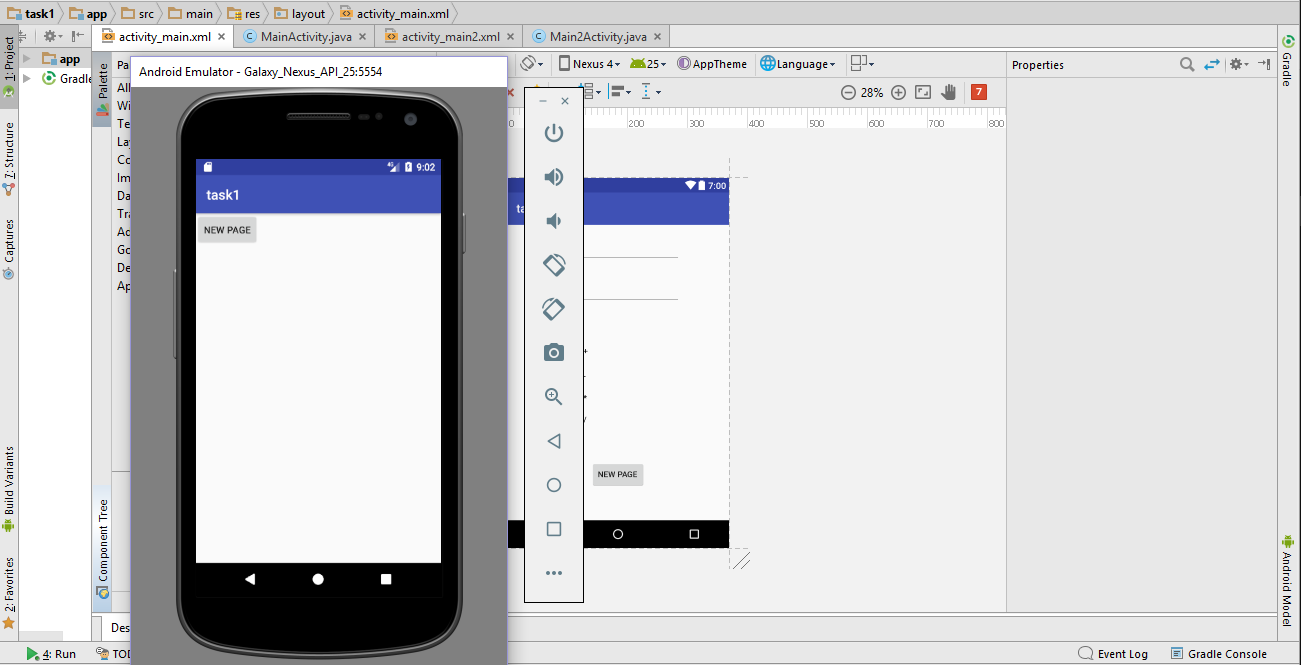
*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.newapp.task1.MainActivity"**>  
 <**EditText  
 android:id="@+id/editText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:onClick="calculation"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.502"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginBottom="8dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/editText2"  
 app:layout\_constraintVertical\_bias="0.402"** />  
 <**EditText  
 android:id="@+id/editText2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:onClick="calculation"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.502"  
 android:layout\_marginBottom="59dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/radioGroup"** />  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="55dp"  
 android:onClick="onClick"  
 android:text="New page"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"** />  
 <**RadioGroup  
 android:id="@+id/radioGroup"  
 android:layout\_width="184dp"  
 android:layout\_height="163dp"  
 android:layout\_marginBottom="48dp"  
 android:weightSum="1"  
 app:layout\_constraintBottom\_toTopOf="@+id/button2"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"**>  
 <**RadioButton  
 android:id="@+id/radioButton2"  
 android:layout\_width="53dp"  
 android:layout\_height="43dp"  
 android:onClick="calculation"  
 android:text="+"** />  
 <**RadioButton  
 android:id="@+id/radioButton"  
 android:layout\_width="53dp"  
 android:layout\_height="44dp"  
 android:onClick="calculation"  
 android:text="-"** />  
 <**RadioButton  
 android:id="@+id/radioButton3"  
 android:layout\_width="53dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.02"  
 android:onClick="calculation"  
 android:text="\*"** />  
 <**RadioButton  
 android:id="@+id/radioButton5"  
 android:layout\_width="55dp"  
 android:layout\_height="43dp"  
 android:onClick="calculation"  
 android:text="/"** />  
 </**RadioGroup**>  
</**android.support.constraint.ConstraintLayout**>

**SecondXML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.newapp.task1.Main2Activity"**>  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onClick"  
 android:text="new page"  
 tools:layout\_editor\_absoluteX="137dp"  
 tools:layout\_editor\_absoluteY="161dp"** />  
</**android.support.constraint.ConstraintLayout**>

**OUTPUT:**





**LAB # 09**

**QUIZ APP**

**CODING:**

**FIRST MAIN ACTIVITY:**

**package** com.example.osama.april62017;

**import** android.content.Intent;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.RadioButton;

**public class** MainActivity **extends** AppCompatActivity {

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

}

**public void** Quiz (View v)

{

RadioButton Khi = (RadioButton)findViewById(R.id.***radioButton4***);

RadioButton Lhr = (RadioButton)findViewById(R.id.***radioButton7***);

RadioButton Isl = (RadioButton)findViewById(R.id.***radioButton6***);

RadioButton Bal = (RadioButton)findViewById(R.id.***radioButton5***);

**if**(Isl.isChecked()){

GlobalVariables.*Score1* = GlobalVariables.*Score1*+ 10;

}

**else if** (Khi.isChecked())

{

GlobalVariables.*Score1*= GlobalVariables.*Score1*+ 0;

}

**else if** (Lhr.isChecked())

{

GlobalVariables.*Score1*= GlobalVariables.*Score1*+ 0;

}

**else if** (Bal.isChecked())

{

GlobalVariables.*Score1*= GlobalVariables.*Score1*+ 0;

}

Intent obj = **new** Intent(MainActivity.**this**,Main2Activity.**class**);

startActivity(obj);

}

}

**SECOND MAIN ACTIVITY:**

**package** com.example.osama.april62017;

**import** android.content.Intent;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.RadioButton;

**public class** Main2Activity **extends** AppCompatActivity {

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main2***);

}

**public void** NextPage (View v)

{

RadioButton Pak = (RadioButton)findViewById(R.id.***radioButton13***);

RadioButton Tes = (RadioButton)findViewById(R.id.***radioButton16***);

RadioButton Apl = (RadioButton)findViewById(R.id.***radioButton15***);

RadioButton Irn = (RadioButton)findViewById(R.id.***radioButton14***);

**if**(Pak.isChecked()){

GlobalVariables.*Score2* = GlobalVariables.*Score2* + 10;

}

**else if** (Apl.isChecked())

{

GlobalVariables.*Score2* = GlobalVariables.*Score2* + 0;

}

**else if** (Irn.isChecked())

{

GlobalVariables.*Score2* = GlobalVariables.*Score2* + 0;

}

**else if** (Tes.isChecked())

{

GlobalVariables.*Score2* = GlobalVariables.*Score2* + 0;

}

Intent obj = **new** Intent(Main2Activity.**this**,Main3Activity.**class**);

startActivity(obj);

}

}

**Third MAIN ACTIVITY:**

**package** com.example.osama.april62017;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.EditText;

**public class** Main3Activity **extends** AppCompatActivity {

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main3***);

}

**public void** result(View v)

{

EditText res = (EditText)findViewById(R.id.***editText***);

res.setText(**"Score is: "**+GlobalVariables.Result);

}

}

**FIRST XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.osama.april62017.MainActivity"**><![CDATA[  
 android:layout\_width="368dp"  
 android:layout\_height="495dp"  
 tools:layout\_editor\_absoluteX="8dp"  
 tools:layout\_editor\_absoluteY="8dp">  
 ]]>  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="300dp"  
 android:layout\_height="90dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="42dp"  
 android:layout\_marginTop="12dp"  
 android:text="Capital of Pakistan ?"  
 android:textAppearance="@style/TextAppearance.AppCompat.Display1"** />  
 <**RadioGroup  
 android:layout\_width="250dp"  
 android:layout\_height="300dp"  
 android:layout\_below="@+id/textView"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="20dp"  
 android:id="@+id/radioGroup2"**>  
 <**RadioButton  
 android:id="@+id/radioButton4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Karachi"** />  
 <**RadioButton  
 android:id="@+id/radioButton7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Lahore"** />  
 <**RadioButton  
 android:id="@+id/radioButton6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Islamabad"** />  
 <**RadioButton  
 android:id="@+id/radioButton5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Balochistan"** />  
 </**RadioGroup**>  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/radioGroup2"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="14dp"  
 android:onClick="Quiz"  
 android:text="NEXT"** />  
</**RelativeLayout**>

**SECOND XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**tools:context="com.example.osama.april62017.Main2Activity"**>

<**RelativeLayout**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**tools:context="com.example.osama.april62017.MainActivity"**><![CDATA[

android:layout\_width="368dp"

android:layout\_height="495dp"

tools:layout\_editor\_absoluteX="8dp"

tools:layout\_editor\_absoluteY="8dp">

]]>

<**RadioGroup**

**android:id="@+id/radioGroup2"**

**android:layout\_width="250dp"**

**android:layout\_height="300dp"**

**android:layout\_below="@+id/textView4"**

**android:layout\_centerHorizontal="true"**

**android:layout\_marginTop="41dp"**>

<**RadioButton**

**android:id="@+id/radioButton13"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder of Pakistan"** />

<**RadioButton**

**android:id="@+id/radioButton16"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder Of Tesla"** />

<**RadioButton**

**android:id="@+id/radioButton15"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder of Apple"** />

<**RadioButton**

**android:id="@+id/radioButton14"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder of Iran"** />

</**RadioGroup**>

<**TextView**

**android:id="@+id/textView4"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:text="Who Is Mohammed Ali Jinnah"**

**android:textAppearance="@style/TextAppearance.AppCompat.Display1"**

**android:layout\_alignParentTop="true"** />

<**Button**

**android:id="@+id/button4"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_alignParentBottom="true"**

**android:layout\_centerHorizontal="true"**

**android:layout\_marginBottom="17dp"**

**android:onClick="NextPage"**

**android:text="Next"** />

</**RelativeLayout**>

</**RelativeLayout**>

**Third XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**tools:context="com.example.osama.april62017.Main3Activity"**>

<**RelativeLayout**

**android:layout\_width="0dp"**

**android:layout\_height="0dp"**

**tools:context="com.example.osama.april62017.Main2Activity"**

**tools:layout\_editor\_absoluteY="8dp"**

**tools:layout\_editor\_absoluteX="8dp"**

**android:id="@+id/relativeLayout"**>

</**RelativeLayout**>

<**RelativeLayout**

**android:layout\_width="0dp"**

**android:layout\_height="0dp"**

**tools:context="com.example.osama.april62017.MainActivity"**

**tools:layout\_editor\_absoluteY="8dp"**

**tools:layout\_editor\_absoluteX="8dp"**><![CDATA[

android:layout\_width="368dp"

android:layout\_height="495dp"

tools:layout\_editor\_absoluteX="8dp"

tools:layout\_editor\_absoluteY="8dp">

]]>

<**RadioGroup**

**android:id="@+id/radioGroup2"**

**android:layout\_width="250dp"**

**android:layout\_height="300dp"**

**android:layout\_below="@+id/textView4"**

**android:layout\_centerHorizontal="true"**

**android:layout\_marginTop="41dp"**>

<**RadioButton**

**android:id="@+id/radioButton13"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder of Pakistan"** />

<**RadioButton**

**android:id="@+id/radioButton16"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder Of Tesla"** />

<**RadioButton**

**android:id="@+id/radioButton15"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder of Apple"** />

<**RadioButton**

**android:id="@+id/radioButton14"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_weight="1"**

**android:text="Founder of Iran"** />

</**RadioGroup**>

<**TextView**

**android:id="@+id/textView4"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_alignParentTop="true"**

**android:text="Who Is Mohammed Ali Jinnah"**

**android:textAppearance="@style/TextAppearance.AppCompat.Display1"** />

<**Button**

**android:id="@+id/button4"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_alignParentBottom="true"**

**android:layout\_centerHorizontal="true"**

**android:layout\_marginBottom="17dp"**

**android:onClick="NextPage"**

**android:text="Next"** />

</**RelativeLayout**>

<**EditText**

**android:id="@+id/editText"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_marginTop="52dp"**

**android:ems="10"**

**android:inputType="textPersonName"**

**android:layout\_below="@+id/textView3"**

**android:layout\_centerHorizontal="true"** />

<**Button**

**android:id="@+id/button3"**

**android:layout\_width="wrap\_content"**

**android:layout\_height="wrap\_content"**

**android:layout\_below="@+id/editText"**

**android:layout\_centerHorizontal="true"**

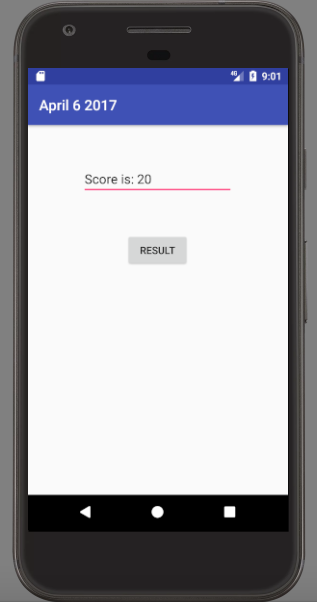
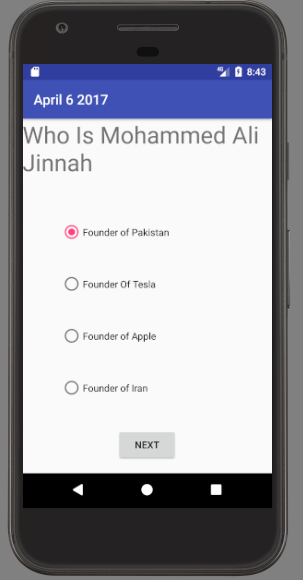
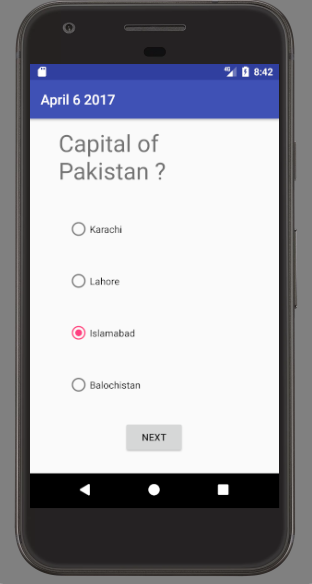
**android:layout\_marginTop="52dp"**

**android:onClick="result"**

**android:text="RESULT"** />

</**RelativeLayout**>

**OUTPUT:**



**LAB # 10**

**QUIZ APP TASK:**

**CODING:**

**MAIN ACTIVITY:**

**package** com.example.abdul.sanaquizapp;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**public class** MainActivity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
 **public void** nextActivity (View view) {  
 Intent intent1 = **new** Intent(MainActivity.**this**, Main2Activity.**class**);  
 startActivity(intent1);  
 }  
}

**MAIN ACTIVITY 2:**

**package** com.example.abdul.sanaquizapp;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.RadioButton;  
**import** android.widget.Toast;  
**public class** Main2Activity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main2***);  
 }  
  
 **public void** checkAnswer(View view) {  
 RadioButton r1 = (RadioButton)findViewById(R.id.***radioButton***);  
 **if** (r1.isChecked())  
 variables.*q1* = **true**;  
 **else** variables.*q1* = **false**;  
 }  
 **public void** nextActivity (View view) {  
 RadioButton r1 = (RadioButton)findViewById(R.id.***radioButton***);  
 RadioButton r2 = (RadioButton)findViewById(R.id.***radioButton2***);  
 RadioButton r3 = (RadioButton)findViewById(R.id.***radioButton3***);  
 **if** (!r1.isChecked() && !r2.isChecked() && !r3.isChecked())  
 (Toast.*makeText*(Main2Activity.**this**,**"Select an option first."**, Toast.***LENGTH\_SHORT***)).show();  
 **else** {  
 Intent intent1 = **new** Intent(Main2Activity.**this**, Main3Activity.**class**);  
 startActivity(intent1);  
 }  
 }  
}

**MAIN ACTIVITY 3:**

**package** com.example.abdul.sanaquizapp;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.RadioButton;  
**import** android.widget.Toast;  
**public class** Main3Activity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main3***);  
 }  
 **public void** checkAnswer(View view) {  
 RadioButton r1 = (RadioButton)findViewById(R.id.***radioButton2***);  
 RadioButton r2 = (RadioButton)findViewById(R.id.***radioButton***);  
 RadioButton r3 = (RadioButton)findViewById(R.id.***radioButton3***);  
 **if** (!r1.isChecked() && !r2.isChecked() && !r3.isChecked())  
 (Toast.*makeText*(Main3Activity.**this**,**"Select an option first!"**,Toast.***LENGTH\_SHORT***)).show();  
 **else** {  
 **if** (r1.isChecked())  
 variables.*q2* = **true**;  
 **else** variables.*q2* = **false**;  
 }  
 }  
 **public void** nextActivity (View view) {  
 Intent intent1 = **new** Intent(Main3Activity.**this**, Main4Activity.**class**);  
 startActivity(intent1);  
 }  
}

**MAIN ACTIVITY 4:**

**package** com.example.abdul.sanaquizapp;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.RadioButton;  
**import** android.widget.Toast;  
**public class** Main4Activity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main4***);  
 }  
 **public void** checkAnswer(View view) {  
 RadioButton r1 = (RadioButton)findViewById(R.id.***radioButton3***);  
 **if** (r1.isChecked())  
 variables.*q3* = **true**;  
 **else** variables.*q3* = **false**;  
 }  
 **public void** nextActivity (View view) {  
 Toast toast=Toast.*makeText*(Main4Activity.**this**,**"Select an option first."**, Toast.***LENGTH\_SHORT***);  
 RadioButton r1 = (RadioButton)findViewById(R.id.***radioButton3***);  
 RadioButton r2 = (RadioButton)findViewById(R.id.***radioButton2***);  
 RadioButton r3 = (RadioButton)findViewById(R.id.***radioButton***);  
 **if** (!r1.isChecked() && !r2.isChecked() && !r3.isChecked()) {  
 **if** (toast != **null**)  
 toast.cancel();  
 toast.show();  
 }  
 **else** {  
 Intent intent1 = **new** Intent(Main4Activity.**this**,Main5Activity.**class**);  
 startActivity(intent1);  
 }  
 }  
}

**MAIN ACTIVITY 5:**

**package** com.example.abdul.sanaquizapp;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.TextView;  
**public class** Main5Activity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main5***);  
 **int** totalCorrectAnswers=0;  
 **if** (variables.*q1*)  
 totalCorrectAnswers++;  
 **if** (variables.*q2*)  
 totalCorrectAnswers++;  
 **if** (variables.*q3*)  
 totalCorrectAnswers++;  
 TextView t = (TextView)findViewById(R.id.***textView5***);  
 t.setText(**""**);  
 t.append(**"Correct answer(s): "**);  
 t.append( Integer.*toString*(totalCorrectAnswers) + **" out of 3"**);  
 }  
 **public void** backToStart(View view) {  
 Intent intent1 = **new** Intent(Main5Activity.**this**, MainActivity.**class**);  
 startActivity(intent1);  
 }  
}

**FIRST XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.abdul.sanaquizapp.MainActivity"**>  
  
  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="nextActivity"  
 android:text="Start"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 android:layout\_marginTop="47dp"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView"**/>  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:text="Start Quiz ?"  
 android:textSize="40sp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 android:layout\_marginTop="35dp"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4"**/>  
 <**TextView  
 android:id="@+id/textView4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="8dp"  
 android:text="This Quiz contains total 3 questions."  
 android:textSize="15sp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**/>  
  
</**android.support.constraint.ConstraintLayout**>

**SECOND XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.abdul.sanaquizapp.Main2Activity"**>  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:onClick="nextActivity"  
 android:text="Next Question"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/radioGroup"**/>  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:text="power of centi?"  
 android:textSize="20sp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3"**/>  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:text="Question 1:"  
 android:textSize="15sp"  
 app:layout\_constraintHorizontal\_bias="0.503"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**/>  
  
 <**RadioGroup  
 android:layout\_width="0dp"  
 android:layout\_height="206dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="0dp"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2"  
 android:id="@+id/radioGroup"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"**>  
  
 <**RadioButton  
 android:id="@+id/radioButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="100"**/>  
  
 <**RadioButton  
 android:id="@+id/radioButton2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="1000"**/>  
  
 <**RadioButton  
 android:id="@+id/radioButton3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="10"**/>  
 </**RadioGroup**>  
  
  
</**android.support.constraint.ConstraintLayout**>

**THIRD XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.abdul.sanaquizapp.Main3Activity"**>  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:onClick="nextActivity"  
 android:text="Next Question"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/radioGroup"**/>  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:text="Which is a Human body part?"  
 android:textSize="20sp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3"**/>  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:text="Question 2:"  
 android:textSize="15sp"  
 app:layout\_constraintHorizontal\_bias="0.503"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**/>  
 <**RadioGroup  
 android:layout\_width="0dp"  
 android:layout\_height="206dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="11dp"  
 app:layout\_constraintHorizontal\_bias="0.501"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2"  
 android:id="@+id/radioGroup"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"**>  
  
 <**RadioButton  
 android:id="@+id/radioButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="Gills"**/>  
  
 <**RadioButton  
 android:id="@+id/radioButton2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="Feet"**/>  
  
 <**RadioButton  
 android:id="@+id/radioButton3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="Tentacles"**/>  
 </**RadioGroup**>  
  
</**android.support.constraint.ConstraintLayout**>

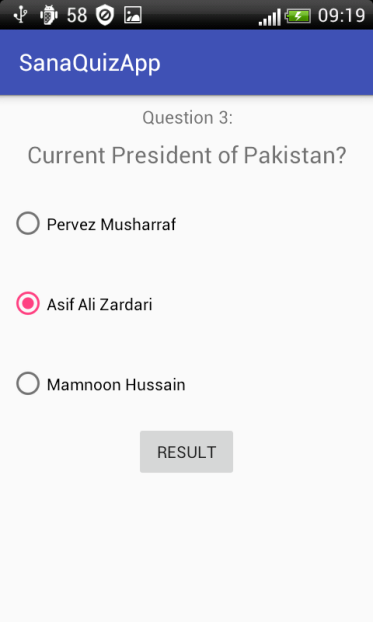
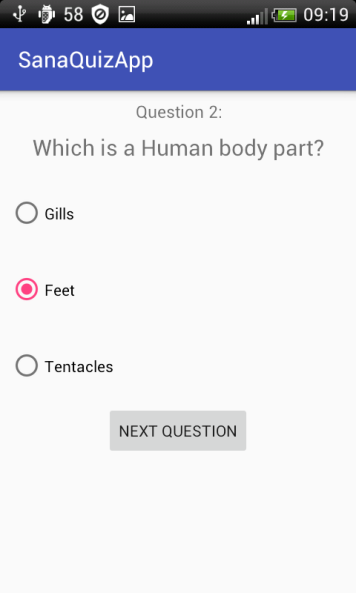
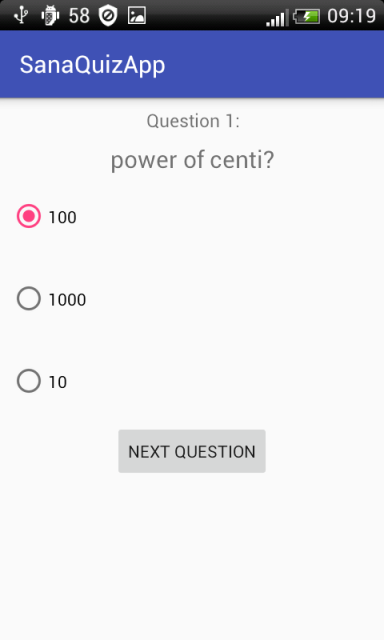
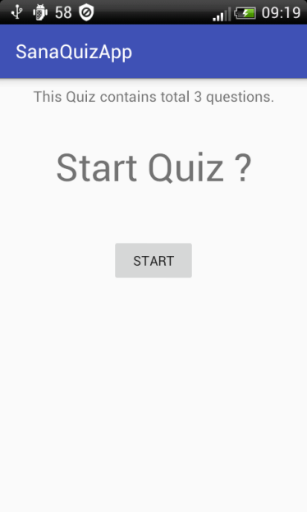
**FOURTH XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.abdul.sanaquizapp.Main4Activity"**>  
  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:onClick="nextActivity"  
 android:text="Result"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/radioGroup"**/>  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:text="Current President of Pakistan?"  
 android:textSize="20sp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3"**/>  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:text="Question 3:"  
 android:textSize="15sp"  
 app:layout\_constraintHorizontal\_bias="0.503"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**/>  
  
 <**RadioGroup  
 android:layout\_width="0dp"  
 android:layout\_height="206dp"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="11dp"  
 app:layout\_constraintHorizontal\_bias="0.501"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView2"  
 android:id="@+id/radioGroup"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"**>  
  
 <**RadioButton  
 android:id="@+id/radioButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="Pervez Musharraf"**/>  
  
 <**RadioButton  
 android:id="@+id/radioButton2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="Asif Ali Zardari"**/>  
  
 <**RadioButton  
 android:id="@+id/radioButton3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:onClick="checkAnswer"  
 android:text="Mamnoon Hussain"**/>  
 </**RadioGroup**>  
</**android.support.constraint.ConstraintLayout**>

**FIFTH XML:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.abdul.sanaquizapp.Main5Activity"**>  
 <**Button  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="150dp"  
 android:onClick="backToStart"  
 android:text="Back to Start page"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView5"  
 app:layout\_constraintHorizontal\_bias="0.502"**/>  
  
 <**TextView  
 android:id="@+id/textView5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="8dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginTop="16dp"  
 android:textSize="20sp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.021"**/>  
  
</**android.support.constraint.ConstraintLayout**>

**OUTPUT:**



**LAB # 11**

**CREATING ACTIVITIES IN ANDROID**

## OBJECTIVES:

1. To become familiar with intent method.
2. Basic concept of activity and multiple activities in android.

## INTRODUCTION TO ACTIVITY:

* Activity is a Java code that supports a screen or UI. In other words, building block of the user interface is the activity.
* Activity class is a pre-defined class in Android and every application, which has UI, must inherit it to create window.
* Let us create a simple application, which displays current time when you just tap the window. We will dissect the code to understand how the activity works.

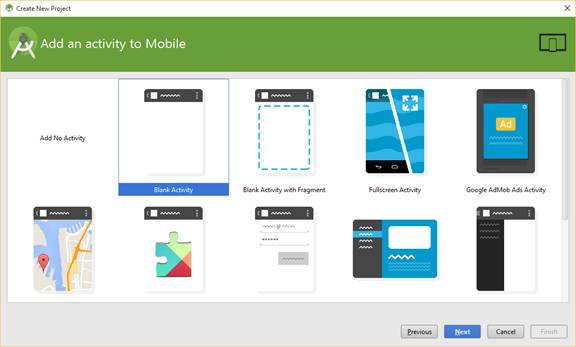
**CONCEPT OF ACTIVITY:**

The [Activity](https://developer.android.com/reference/android/app/Activity.html) class serves as the entry point for an app’s interaction with the user, providing the window in which the app draws its UI. This window typically fills the screen, but may be smaller than the screen and float on top of other windows. You implement an activity as a subclass of the [Activity](https://developer.android.com/reference/android/app/Activity.html) class. Generally, one activity implements one screen in an app. For instance, one of an app’s activities may implement the Preferences screen, while another activity implements the Compose Email screen.

Most apps contain multiple screens, which means they comprise multiple activities. Typically, one activity in an app is specified as the *main activity*, which is the first screen to appear when the user launches the app. Each activity can then start another activity to perform different actions. For example, the main activity in a simple e-mail app may provide the screen that shows an e-mail inbox. From there, the main activity might launch other activities that provide screens for tasks like writing e-mails and opening individual e-mails.

Although activities work together to form a cohesive user experience in an app, each activity is only loosely bound to the other activities, there are usually minimal dependencies among the activities in an app. In fact, activities often start up activities belonging to other apps. For example, a browser app might launch the Share activity of a social-media app.

**EXAMPLE OF MAIN ACTIVITY:**

****

**MULTIPLE ACTIVITIES:**

Even the simplest applications have more than one functionality. Hence, there is often a need to deal with multiple activities. For example, a game can have two activities: a high scores screen and a game screen. A notepad can have three activities: view a list of notes, read a selected note, and edit a selected or new note.

The main activity, as defined in the Android Manifest XML file, is started when the application is started. This activity can launch another activity, usually after a trigger event. This causes the main activity to pause while the secondary activity is active. When the secondary activity ends, the main activity is brought to the foreground and resumed.

There is a one method for switching one activity to another is called **intent method.**

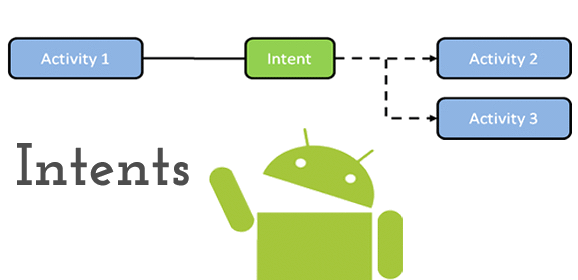
**INTENT METHOD:**

Android application components can connect to other Android applications. This connection is based on a task description represented by an Intent object.

*Intents* are asynchronous messages, which allow application components to request functionality from other Android components. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. For example, an activity can start an external activity for taking a picture.

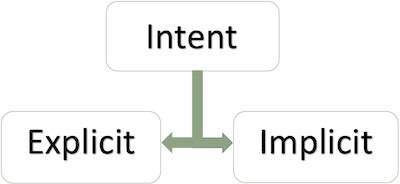
Intents are objects of the android. Content. Intent type. Your code can send them to the Android system defining the components you are targeting. For example, via the startActivity() method you can define that the intent should be used to start an activity.

An intent can contain data via a Bundle. This data can be used by the receiving component.



**TYPES OF INTENTS:**

There are following two types of intents supported by Android



### Explicit Intents:

Explicit intent going to be connected internal world of application, suppose if you want to connect one activity to another activity, we can do this quote by explicit intent, below image is connecting first activity to second activity by clicking button.

These intents designate the target component by its name and they are typically used for application-internal messages - such as an activity starting a subordinate service or launching a sister activity. For example −

// Explicit Intent by specifying its class name

Intent i = new Intent (FirstActivity.this, SecondActivity.class);

// Starts TargetActivity

startActivity(i);

### Implicit Intents:

These intents do not name a target and the field for the component name is left blank. Implicit intents are often used to activate components in other applications. For example −

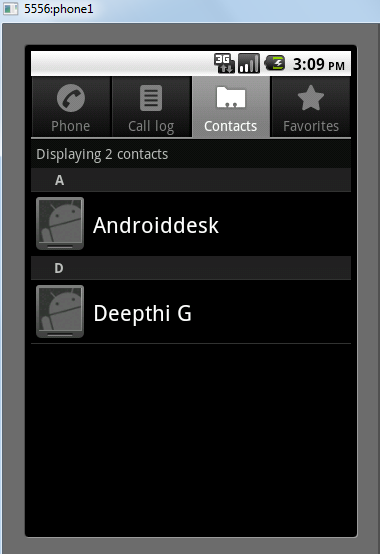
Intent read1=new Intent();

read1.setAction(android.content.Intent.ACTION\_VIEW);

read1.setData(ContactsContract.Contacts.CONTENT\_URI);

startActivity(read1);

Above code will give result as shown below:



The target component which receives the intent can use the **getExtras()** method to get the extra data sent by the source component. For example −

// Get bundle object at appropriate place in your code

Bundle extras = getIntent().getExtras();

// Extract data using passed keys

String value1 = extras.getString("Key1");

String value2 = extras.getString("Key2");

## PROCEDURE FOR CREATING MULTIPLE ACTIVITIES:

1. Create a new project **File -> Android Project**. While creating a new project give activity name as **FirstScreenActivity**.
2. Now you need to create user interface for the FirstScreenActivity.java
3. Create a new xml file in layout folder or rename the main.xml to screen1.xml  
   **Right Click on Layout -> New -> Android XML file** and name it as **screen1.xml**
4. Now insert the following code in screen1.xml to design a small layout. This layout contains simple form with a button.

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

    android:orientation="vertical"

    android:layout\_width="fill\_parent"

    android:layout\_height="fill\_parent"

    >

    <TextView android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:text="Name: "/>

    <EditText android:id="@+id/name"

            android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_marginBottom="10dip"/>

    <TextView

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:text="Email: "

        />

    <EditText android:id="@+id/email"

            android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_marginBottom="10dip"/>

    <Button android:id="@+id/btnNextScreen"

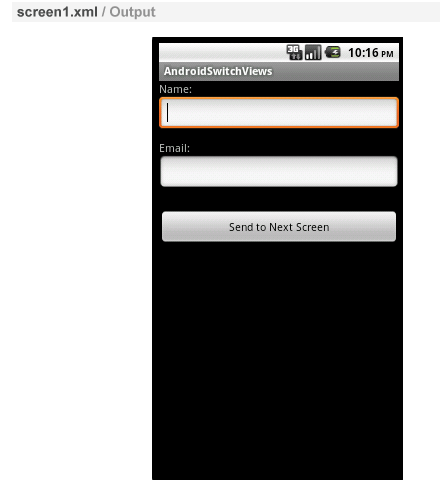
            android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:text="Send to Next Screen"

            android:layout\_marginTop="15dip"/>

</LinearLayout>



1. Now open your **FirstScreenActivity.java** and Type the following code. In the following code we are creating a new Intent and passing parameters on clicking button.

**FIRST MAIN ACTIVITY:**

package com.example.androidswitchviews;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

public class FirstScreenActivity extends Activity {

    // Initializing variables

    EditText inputName;

    EditText inputEmail;

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.screen1);

        inputName = (EditText) findViewById(R.id.name);

        inputEmail = (EditText) findViewById(R.id.email);

        Button btnNextScreen = (Button) findViewById(R.id.btnNextScreen);

        //Listening to button event

        btnNextScreen.setOnClickListener(new View.OnClickListener() {

            public void onClick(View arg0) {

                //Starting a new Intent

                Intent nextScreen = new Intent(getApplicationContext(), SecondScreenActivity.class);

                //Sending data to another Activity

                nextScreen.putExtra("name", inputName.getText().toString());

                nextScreen.putExtra("email", inputEmail.getText().toString());

                Log.e("n", inputName.getText()+"."+ inputEmail.getText());

                startActivity(nextScreen);

            }

        });

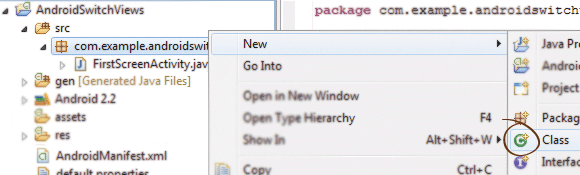
    }

}

|  |
| --- |
|  |

1. Create a class called **SecondScreenActivity.java**. **Right Click on**

**src/yourpackagefolder -> New -> Class** and name it as **SecondScreenActivity.java**



1. Now we need interface for our Second Actvity. Create a new xml file and name it as screen2.xml.  
   **Right Click on Layout -> New -> Android XML file** and name it as **screen2.xml**. Insert the following code in screen2.xml.

**XML:**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

  xmlns:android="<http://schemas.android.com/apk/res/android>"

  android:orientation="vertical"

  android:layout\_width="match\_parent"

  android:layout\_height="match\_parent">

  <TextView android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:text="You Entered..."

            android:textSize="25dip"

            android:gravity="center"

            android:layout\_margin="15dip"/>

  <TextView android:id="@+id/txtName"

            android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_margin="15dip"

            android:textSize="18dip"/>

  <TextView android:id="@+id/txtEmail"

            android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_margin="15dip"

            android:textSize="18dip"/>

  <Button android:id="@+id/btnClose"

            android:layout\_width="fill\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_marginTop="15dip"

            android:text="Close"/>

</LinearLayout>

|  |
| --- |
|  |

1. Now open **SecondScreenActivity.java** and type the following code. Here we are simply reading the parameters and displaying them on to screen.

**SECOND MAIN ACTIVITY:**

package com.example.androidswitchviews;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class SecondScreenActivity extends Activity {

    /\*\* Called when the activity is first created. \*/

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.screen2);

        TextView txtName = (TextView) findViewById(R.id.txtName);

        TextView txtEmail = (TextView) findViewById(R.id.txtEmail);

        Button btnClose = (Button) findViewById(R.id.btnClose);

        Intent i = getIntent();

        // Receiving the Data

        String name = i.getStringExtra("name");

        String email = i.getStringExtra("email");

        Log.e("Second Screen", name + "." + email);

        // Displaying Received data

        txtName.setText(name);

        txtEmail.setText(email);

        // Binding Click event to Button

        btnClose.setOnClickListener(new View.OnClickListener() {

            public void onClick(View arg0) {

                //Closing SecondScreen Activity

                finish();

            }

        });

    }}

|  |
| --- |
|  |

1. Now everything is ready and before running your project make sure that you an entry of new activity name in **AndroidManifest.xml** file. Open you AndroidManifest.xml file and modify the code as below:

**SECOND XML:**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="<http://schemas.android.com/apk/res/android>"

      package="com.example.androidswitchviews"

      android:versionCode="1"

      android:versionName="1.0">

      <uses-sdk android:minSdkVersion="8" />

    <application android:icon="@drawable/icon" android:label="@string/app\_name">

        <activity android:name=".FirstScreenActivity"

                  android:label="@string/app\_name">

            <intent-filter>

                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />

            </intent-filter>

        </activity>

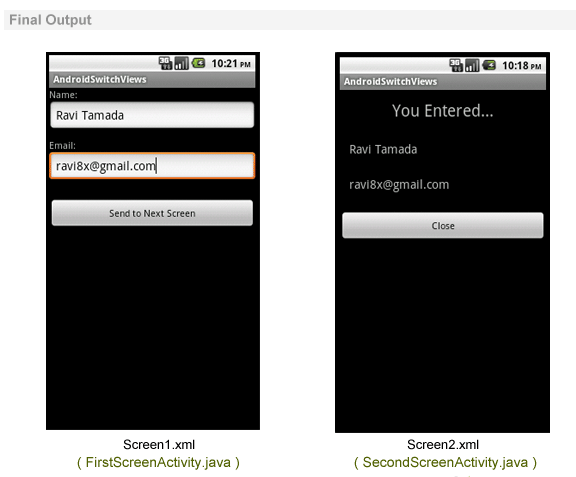
        <!-- Add new Activity class name here --->

        <activity android:name=".SecondScreen"></activity>

    </application>

</manifest>

1. Finally run your project by **right clicking on your project folder -> Run As -> 1 Android Application**. You can see the application is running by switching between screens. The below image is output screenshots of both xml files.



**LAB # 12**

**INTRODUCTION TO DATABASE IN ANDROID**

## OBJECTIVES:

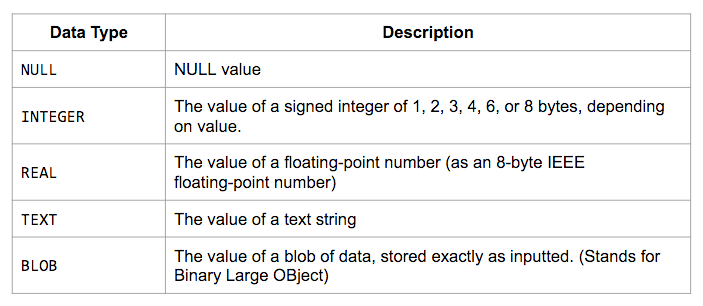
1. To become familiar with android database.
2. Basic concept of SQL lite.

**INTRODUCTION TO SQLite DATABASES:**

SQLite is the main database backend that Android uses to manage complicated data. It provides more structure and functionality than storing data in a file or anywhere else. Database backends are widely used to store data locally on the device and sync with a web server, for example. But before we can do this, we need to learn more about databases and SQLite, which defines its own “programming language” and system. We’ll learn about what databases are and how to do basic CRUD (Create, Read, Update, and Delete) operations with them in SQLite.

**DATA TYPES USED IN SQLITE:**

Now that we have a blank database, we can create a table with a very specific schema. But before we do that, we need more information on what the data types for SQLite columns. They are enumerated in the table below:

[](https://androidkennel.org/wp-content/uploads/2015/11/Intro-to-DBs-%E2%80%93-3.png?x65546)

We will only be using INTEGER, REAL and TEXT since they are the most common, but NULL and BLOB exist as well. The data types store exactly what you think they store based on their names:

**CREATE:**

Now that we have a table, let us actually add some rows to it! To insert rows, we have to begin a transaction since we are modifying the database. We can query all we want, but if we are adding new rows, updating existing rows, or deleting rows, we need to use a transaction to do so. All transactions begin with BEGIN; and end with COMMIT; and we will be wrapping our entire potential database changes in these markers like so. Execute the following SQLite commands.

|  |  |
| --- | --- |
| 1  2  3 | BEGIN;  INSERT INTO student(\_id, name, age, gPA) VALUES(1, 'John Doe', 18, 3.81);  COMMIT; |

Let us talk about the insert statement. First, we have to say what table we are inserting rows into! Then comes a comma-delimited list of columns in parentheses that we want to insert into and then the values that we want to insert into those columns. After this, we should have our row inserted into the database! We will learn how to read it later, but let us add a few more rows and go over different styles of insert statements.

The purpose of the above syntax is to specify that we are going to be inserting exactly these values into these columns of the table. If we are going to provide values for each column, we can omit the comma-separated list of columns. Execute the following statement.

|  |  |
| --- | --- |
| 1  2  3 | BEGIN;  INSERT INTO student VALUES(2, 'Jane Doe', 20, 3.57);  COMMIT; |

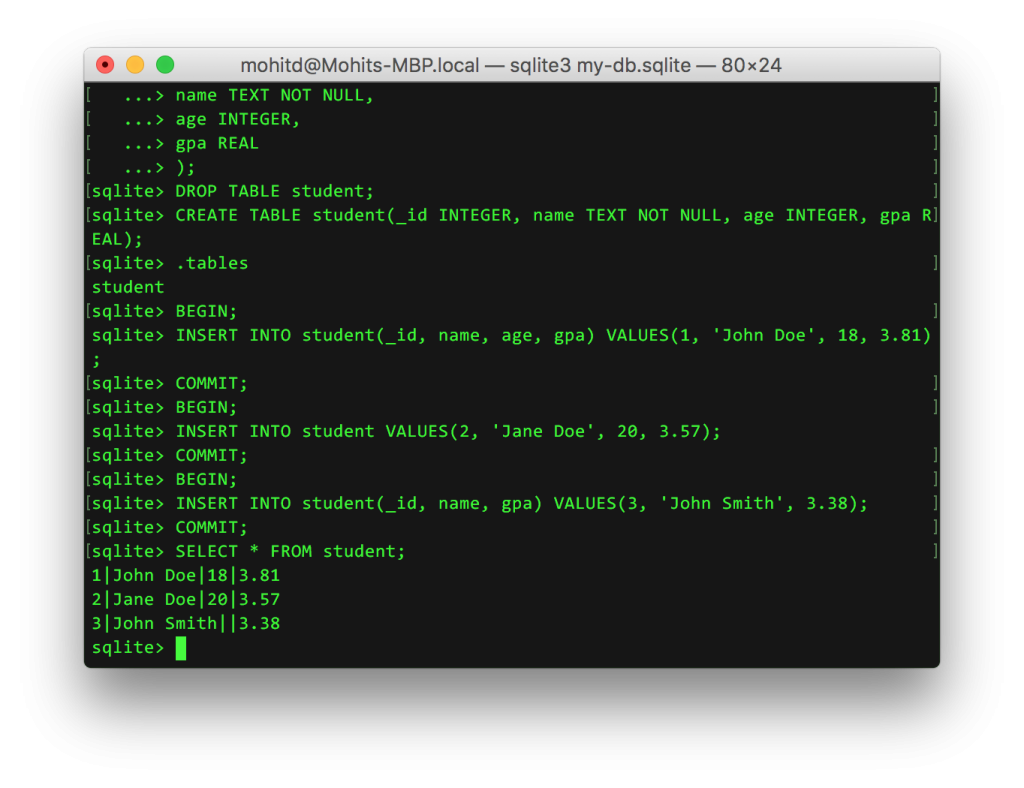
We can see that the columns are omitted and SQLite associates the first value with the first column and so on. Therefore, the ordering of the columns in the table is important! It allows us to do this kind of shorthand. In addition to this method of inserting, we can insert only values into rows. Suppose we wanted to omit one’s age for privacy reasons. If that column is not marked with the NOT NULL attribute, we can do something like the following. Execute the following statement:

|  |  |
| --- | --- |
| 1  2  3 | BEGIN;  INSERT INTO student(\_id, name, gpa) VALUES(3, 'John Smith', 3.38);  COMMIT; |

We can see that we omitted the age column so this row will not have an age if we were to query for it. Note that the age is not simply zero; the age does not exist! Now we have three rows in our database! Let us learn how to access these columns next.

**READ:**

Now that we have some data, let us see how we can query for it. Let us first retrieve all of rows in our only table. Note that we are not actually modifying the database, so we will not need to use a transaction. Instead, we can use a select statement. Execute the following statement: SELECT \*FROM student; we should see three rows where the columns are delimited by a pipe character |.

[](https://androidkennel.org/wp-content/uploads/2015/11/Intro-to-DBs-%E2%80%93-4.png?x65546)

The star means to get all of the columns. We can choose to only retrieve specific columns by replacing the asterisk in the select statement with comma-delimited columns names like SELECT name FROM student; and we can only get all of the names. Now suppose we want to retrieve rows that fit a criterion. We can add a where clause and search for more specific data like the following query that only retrieves student names who have a GPA of 3.5 or higher: SELECT name FROM student WHERE gpa >= 3.5;

This only scratches the surface of the kinds of powerful queries we can do with databases!

### UPDATE:

Suppose we want to make a change to an existing row. We can do that by using the update statement. For example, suppose John Smith did very well in his classes so we need to update his GPA. We can do that using the following.

|  |  |
| --- | --- |
| 1  2  3 | BEGIN;  UPDATE student SET gpa=3.52 WHERE \_id=3;  COMMIT; |

Note that this **does**change the database so we needed to use a transaction. You can see we use the unique id of John to change his GPA. We could also have used the condition where name was John Smith, or, better yet, we could use wildcard characters, but we will not get into those in this post. After the SET keyword, we could provide any number of columns to change, delimited by commas. Any columns not in the list will not be changed.

### DELETE:

Suppose we want to delete a row from our database. This is going to look similar to update, except we will not need any values. Suppose John Smith transferred to another school and we need to remove him from our database.

|  |  |
| --- | --- |
| 1  2  3 | BEGIN;  DELETE FROM student WHERE \_id=3;  COMMIT; |

The delete statement is straightforward and will remove any row from our database that satisfies the where clause.

### ADDITIONAL PROPERTIES:

It is common to have an ID column where each row can be identified uniquely. We have been managing it ourselves, but we can have SQLite do this for us. This happens at database creation though, so we will also learn how to safely transfer data between tables so we do not lose all our users’ data! First, we will rename the old table and create the new one. You will notice that we now use the additional attributes PRIMARY KEY and AUTOINCREMENT. The former means that this column holds unique values for each row and the latter means to automatically increment this column if it is not given a specific value in the insert statement. Then we copy over all our data and delete the old table. All of this looks like the following:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | BEGIN;  ALTER TABLE student RENAME TO tmp\_student;  CREATE TABLE student (\_id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, age INTEGER, gpa REAL);  INSERT INTO student SELECT \* FROM tmp\_student;  DROP TABLE tmp\_student;  COMMIT; |

Now we can use insert statements like INSERT INTO student (name, age, gpa)VALUES('Michael Smith', 20, 3.77)  and there will be a new row with an ID that’s one greater than the previous row, even if that row is deleted! We are **guaranteed** unique values for rows and PRIMARY KEY helps to enforce that as well.

This technique can be used whenever we want to change our table schema drastically. We can use ALTER TABLE to rename our old table, create a new one with a new schema, copy over all of our old data, and remove the old table. This method is safe, secure, and prevents your users from losing all their data!

## CONCLUSION:

We learned about relational databases (RDBs) and SQLite. We covered the data types of the language and how to create tables. We then learned the basic create, read, update, and delete (CRUD) operations that we can perform. Then we finished with some basic techniques and additional modifiers. SQLite is the backend that Android’s Content Providers use to store data and they can be synced with data from web servers or other providers.

**LAB # 13**

**SQLITE FEATURES**

## OBJECTIVES:

1. To become familiar with SQLITE Features.
2. Uses for SQLite.

## SQLite Features:

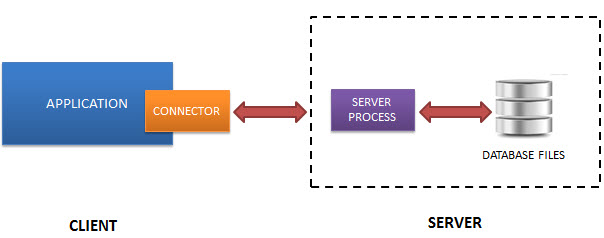
SQLite is a software library that provides a relational database management system. The lite in SQLite means lightweight in terms of setup, database administration, and required resource.

SQLite has the following noticeable features: self-contained, server less, zero-configuration, transactional.

### Server less:

Normally, an RDBMS such as MySQL, PostgreSQL, etc., requires a separate server process to operate. The applications that want to access the database server use TCP/IP protocol to send and receive requests. This is called client/server architecture.

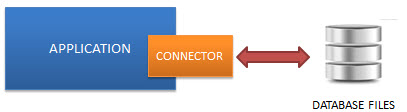
The following diagram illustrates the RDBMS client/server architecture:



* SQLite does NOT work this way.
* SQLite does NOT require a server to run.

SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write directly from the database files stored on disk.

The following diagram illustrates the SQLite server-less architecture:



### Self-Contained:

SQLite is self-contained means it requires minimal support from the operating system or external library. This makes SQLite usable in any environments especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc.

SQLite is developed using ANSI-C. The source code is available as a big sqlite3.c and its header file sqlite3.h. If you want to develop an application that uses SQLite, you just need to drop these files into your project and compile it with your code.

### Zero-configuration:

Because of the server less architecture, you do not need to “install” SQLite before using it. There is no server process that needs to be configured, started, and stopped.

In addition, SQLite does not use any configuration files.

### Transactional:

All transactions in SQLite are fully ACID-compliant. It means all queries and changes are Atomic, Consistent, Isolated, and Durable.

In other words, all changes within a transaction take place completely or not at all even when an unexpected situation like application crash, power failure, or operating system crash occurs.

SQLite is the main database backend that Android uses to manage complicated data. It provides more structure and functionality than storing data in a file or anywhere else. Database backend are widely used to store data locally on the device and sync with a web server, for example. However, before we can do this, we need to learn more about databases and SQLite, which defines its own “programming language” and system. We will learn about what databases are and how to do basic CRUD (Create, Read, Update, and Delete) operations with them in SQLite.

## Uses for SQLite:

* Database for The Internet of Things:SQLite is popular choice for the database engine in cellphones, PDAs, MP3 players, set-top boxes, and other electronic gadgets. SQLite has a small code footprint, makes

Efficient use of memory, disk space, and disk bandwidth, is highly reliable, and requires no maintenance from a Database Administrator.

* Application File Format:Rather than using open () to write XML, JSON, CSV, or some proprietary format into disk files used by your application, use a SQLite database. You will avoid having to write and troubleshoot a parser, your data will be more easily accessible and cross-platform, and your updates will be transactional.
* Website Database:Because it requires no configuration and stores information in ordinary disk files, SQLite is a popular choice as the database to back small to medium-sized websites.
* Stand-in For An Enterprise RDBMS:SQLite is often used as a surrogate for an enterprise RDBMS for demonstration purposes or for testing. SQLite is fast and requires no setup, which takes a lot of the hassle out of testing and which makes demos perky and easy to launch.