

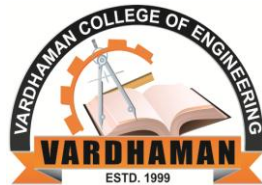
**LAB MANUAL**  
**MOBILE APPLICATION DEVELOPMENT LAB**  
**B. Tech III YEAR I SEMESTER**  
**VCE – R19**

**Mrs G Sahithi**

Assistant Professor  
Department of CSE  
&

**Mr D Ganesh**

Assistant Professor  
Department of CSE



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VARDHAMAN COLLEGE OF ENGINEERING**

**(Autonomous)**

Shamshabad – 501 218, Hyderabad

# **MOBILE APPLICATION DEVELOPMENT**

**COURSE CODE: A5606**

## **COURSE OUTCOMES:**

1. Install and configure Android application development tools, Apply Java programming concepts to Android application development
2. Design and develop user Interfaces for the Android platform
3. Understand the technical challenges posed by current mobile devices and wireless communications; be able to evaluate and select appropriate solutions
4. Select and evaluate suitable software tools and APIs for the development of a particular mobile application and understand their strengths, scope and limitations
5. The students will be able to develop mobile applications with underlying database supports

## LIST OF EXPERIMENTS

S.No	Experiment	Course Outcome
1a	Create an android app to illustrate activity life cycle	CO1
1b	App to visit a specified web page using Implicit Intent	CO2
1c	Implementation of Explicit Intents	CO2
2a	App to perform Mathematical operations(+, -, *, /)	CO1
2b	Text Styles and alignment using checkboxes/Radio Buttons	CO2
3a	Implementation of Spinner Control	CO2
3b	Calculate age of a person using date picker	CO2
4	An Android app for login control and validate details	CO4
5a	App to demonstrate AlertDialog	CO2
5b	App to demonstrate WebView	CO2
6a	App to show Analog and Digital Clocks	CO2
6b	Android app to illustrate Progress bar	CO2
7	Android app to demonstrate Option menu	CO2
8	App to demonstrate Image Switcher	CO2
9a	App to demonstrate Media Player	CO4
9b	Demonstration of Telephony Manager(Phone Contacts)	CO5
10	Implementation of Camera	CO4
11a	Demonstration of sending e-mail	CO3
11b	Demonstration of sending SMS	CO3
12	An Android app to demonstrate database operations using SQLite (insert,Deletion,Update,View,View all)	CO5

## MOBILE APPLICATION DEVELOPMENT LAB

**COURSE CODE: A5606**

**COURSE OUTCOMES:**

1. Install and configure Android application development tools, Apply Java programming concepts to Android application development
2. Design and develop user Interfaces for the Android platform
3. Understand the technical challenges posed by current mobile devices and wireless communications; be able to evaluate and select appropriate solutions
4. Select and evaluate suitable software tools and APIs for the development of a particular mobile application and understand their strengths, scope and limitations
5. The students will be able to develop mobile applications with underlying database supports

**CO-PO MAPPING:**

[illegible]

## JUSTIFICATION FOR CO-PO MAPPING

MAPPING	JUSTIFICATION
CO1-PO5	As CO1 refers to usage of application development tools, it maps with PO5
CO2-PO3	As CO2 refers to design and development of solutions, it matches with PO3 which refers to design and development of solutions.
CO2-PSO1	As CO1 refers to development of user interfaces which involves collection of requirements, analysis, design and test phases, It maps to PSO1
CO3-PO2	As analysis is required to evaluate and select appropriate solution, it maps to PO2
CO4-PO5	Selection of appropriate modern tool involves and hence it maps with PO5
CO5-PO3	Students develop solutions for given problem domain which matches with PO3
CO5-PSO1	Development of solution involves the phases of requirements gathering, analysis, design and testing phases, CO5 maps with PSO1 also.

## WEEK-1

**1a) Aim: Create an android app to illustrate activity life cycle.**

### activity\_main.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="example.myapplication.com.activitylifecycle.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

### MainActivity.java

```
package example.myapplication.com.activitylifecycle;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import android.util.Log;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

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

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        Log.d("lifecycle", "onCreate invoked");
```

```
    }
```

```
    @Override
```

```
    protected void onStart() {
```

```

super.onStart();
Log.d("lifecycle","onStart invoked");
}

@Override
protected void onResume() {
super.onResume();
Log.d("lifecycle","onResume invoked");
}
@Override
protected void onPause() {
super.onPause();
Log.d("lifecycle","onPause invoked");
}
@Override
protected void onStop() {
super.onStop();
Log.d("lifecycle","onStop invoked");
}
@Override
protected void onRestart() {
super.onRestart();
Log.d("lifecycle","onRestart invoked");
}
@Override
protected void onDestroy() {
super.onDestroy();
Log.d("lifecycle","onDestroy invoked");
}
}

```

#### **Prelab VIVA Questions:**

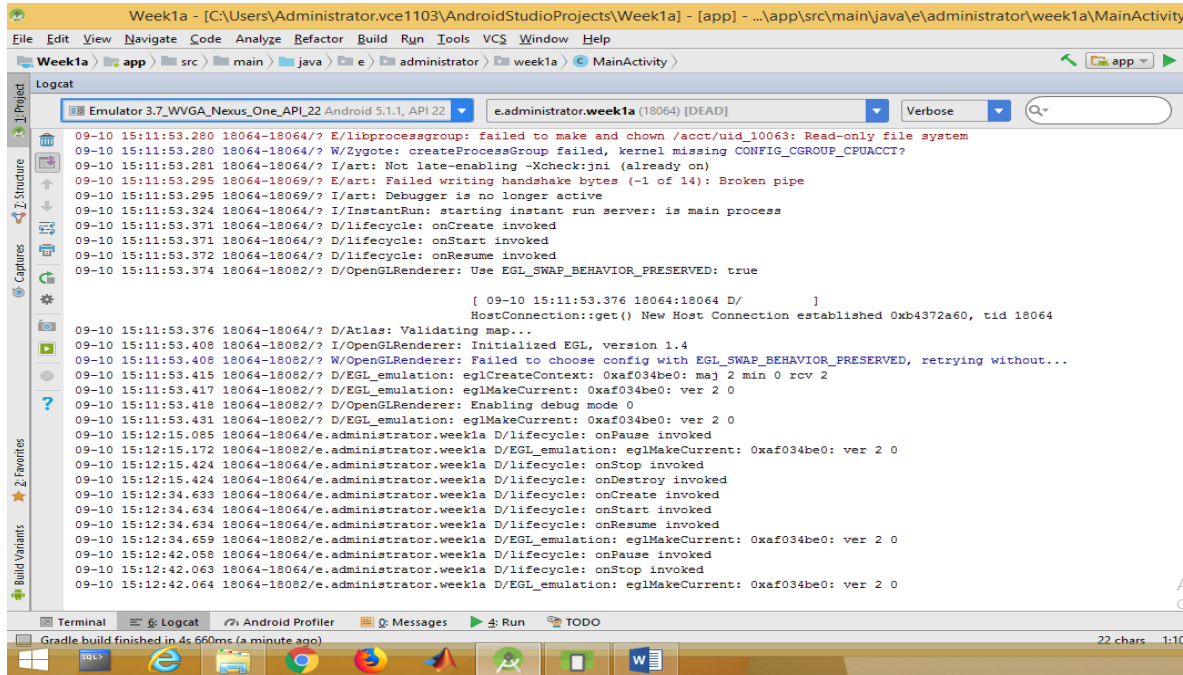
1. What is Activity?
2. What are different life cycle phases of Activity?
3. What is the use of TextView component?
4. What are the different attributes of TextView Component?
5. What is the syntax of Button Component?
6. State the different attributes of Button Component and write Button component XML tag.
7. What does AndroidManifest.XML file contains?
8. Write about Logcat?

#### **Post Lab VIVA Questions :**

1. Write the code snippet of default activity onCreate().
2. What is method to be used to set the layout file to be displayed?
3. What are the different packages and classes to be imported for this program?

4. When Restart() method will be called by an activity?
5. When start() method will be called by an activity?

## Output:



The screenshot shows the Android Studio interface with the Logcat window open. The Logcat displays a series of log messages from the application 'e.administrator.week1a' (PID 18064). The messages include system-level errors, warnings, and lifecycle events. The application appears to have started successfully, with lifecycle methods like onCreate, onStart, onResume, onPause, and onStop being invoked. The Logcat also shows a warning about the EGL configuration and a message about the OpenGLRenderer using the EGL\_SWAP\_BEHAVIOR\_PRESERVED flag.

```
09-10 15:11:53.280 18064-18064/? E/libprocessgroup: failed to make and chown /acct/uid_10063: Read-only file system
09-10 15:11:53.280 18064-18064/? W/Zygote: createProcessGroup failed, kernel missing CONFIG_CGROUP_CPUACCT?
09-10 15:11:53.281 18064-18064/? I/art: Not late-enabling -Xcheck:jni (already on)
09-10 15:11:53.295 18064-18069/? E/art: Failed writing handshake bytes (-1 of 14): Broken pipe
09-10 15:11:53.295 18064-18069/? I/art: Debugger is no longer active
09-10 15:11:53.324 18064-18064/? I/InstantRun: starting instant run server: is main process
09-10 15:11:53.371 18064-18064/? D/lifecycle: onCreate invoked
09-10 15:11:53.371 18064-18064/? D/lifecycle: onStart invoked
09-10 15:11:53.372 18064-18064/? D/lifecycle: onResume invoked
09-10 15:11:53.374 18064-18062/? D/OpenGLRenderer: Use EGL_SWAP_BEHAVIOR_PRESERVED: true
[ 09-10 15:11:53.376 18064:18064 D/
HostConnection::get() New Host Connection established 0xb4372a60, tid 18064
09-10 15:11:53.376 18064-18064/? D/Atlas: Validating map...
09-10 15:11:53.408 18064-18062/? I/OpenGLRenderer: Initialized EGL, version 1.4
09-10 15:11:53.408 18064-18062/? W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
09-10 15:11:53.415 18064-18062/? D/EGL_emulation: eglCreateContext: 0xaf034be0: maj 2 min 0 rcv 2
09-10 15:11:53.417 18064-18062/? D/EGL_emulation: eglMakeCurrent: 0xaf034be0: ver 2 0
09-10 15:11:53.418 18064-18062/? D/OpenGLRenderer: Enabling debug mode 0
09-10 15:11:53.431 18064-18062/? D/EGL_emulation: eglMakeCurrent: 0xaf034be0: ver 2 0
09-10 15:12:15.085 18064-18064/e.administrator.week1a D/lifecycle: onPause invoked
09-10 15:12:15.172 18064-18062/e.administrator.week1a D/EGL_emulation: eglMakeCurrent: 0xaf034be0: ver 2 0
09-10 15:12:15.424 18064-18064/e.administrator.week1a D/lifecycle: onStop invoked
09-10 15:12:15.424 18064-18064/e.administrator.week1a D/lifecycle: onDestroy invoked
09-10 15:12:34.633 18064-18064/e.administrator.week1a D/lifecycle: onCreate invoked
09-10 15:12:34.634 18064-18064/e.administrator.week1a D/lifecycle: onStart invoked
09-10 15:12:34.634 18064-18064/e.administrator.week1a D/lifecycle: onResume invoked
09-10 15:12:34.659 18064-18062/e.administrator.week1a D/EGL_emulation: eglMakeCurrent: 0xaf034be0: ver 2 0
09-10 15:12:42.058 18064-18064/e.administrator.week1a D/lifecycle: onPause invoked
09-10 15:12:42.063 18064-18064/e.administrator.week1a D/lifecycle: onStop invoked
09-10 15:12:42.064 18064-18062/e.administrator.week1a D/EGL_emulation: eglMakeCurrent: 0xaf034be0: ver 2 0
```



**1b) Aim: Create an android app to visit a specified webpage (Use Implicit Intent).**

**activity\_main.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="example.myapplication.com.implicitintent.MainActivity">
```

**<EditText**

```
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="60dp"
    android:ems="10"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.575"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

**<Button**

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="8dp"
    android:layout_marginLeft="156dp"
    android:layout_marginTop="172dp"
    android:text="Visit"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />
</android.support.constraint.ConstraintLayout>
```

**MainActivity.java**

```
package example.myapplication.com.implicitintent;
```

```
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    Button button;
    EditText editText;
```

```
    @Override
```

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

```
        button = findViewById(R.id.button);
        editText = findViewById(R.id.editText);
```

```
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String url=editText.getText().toString();
                Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(intent);
            }
        });
    }
}
```

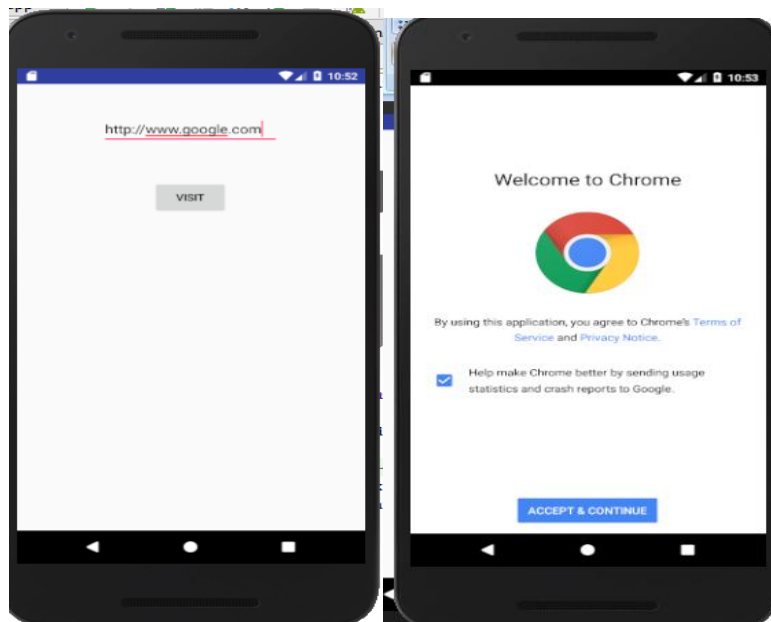
#### **Pre Lab VIVA Questions:**

1. What is intent?
2. What is implicit intent?
3. State the difference between implicit and explicit intent.
4. What is Activity?
5. What are different life cycle phases of Activity?
6. What is the use of TextView component?
7. What are the different attributes of TextView Component?
8. What is the syntax of Button Component?
9. State the different attributes of Button Component and write Button component XML tag.
10. What does AndroidManifest.XML file contains?

### Post Lab VIVA Questions :

1. Write the Listener interface to be implemented for event handling on button click.
2. What is the class to be used for creating an intent?
3. Implicit intent refers to calling an activity of other application or same application?
4. Write the code snippet of default activity onCreate().
5. What is method to be used to set the layout file to be displayed?

### Output :



**1c) Aim: Create an android app to navigate between activities (Use Explicit Intent).**

**activity\_main.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="example.myapplication.com.explicitintent.FirstActivity">
```

**<TextView**

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="First Activity"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.454"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.06" />
```

**<Button**

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="392dp"
    android:onClick="callSecondActivity"
    android:text="Call second activity"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

</android.support.constraint.ConstraintLayout>

### MainActivityOne.java

```
package example.myapplication.com.explicitintent;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
public class FirstActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
    }
    public void callSecondActivity(View view){
        Intent i = new Intent(getApplicationContext(), SecondActivity.class);
        i.putExtra("Value1", "Android By Myapplication");
        i.putExtra("Value2", "Simple ");
        // Set the request code to any code you like, you can identify the
        // callback via this code
        startActivity(i);
    }
}
```

### activitytwo\_main.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="example.myapplication.com.explicitintent.SecondActivity">
```

#### **<TextView**

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="Second Activity"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.454"
```

```

app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.06" />

```

#### **<Button**

```

android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"
android:layout_marginTop="392dp"
android:onClick="callFirstActivity"
android:text="Call first activity"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>

```

ActivityTwo class

#### ***MainActivityTwo.java***

```

package example.myapplication.com.explicitintent;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        Bundle extras = getIntent().getExtras();
        String value1 = extras.getString("Value1");
        String value2 = extras.getString("Value2");
        Toast.makeText(getApplicationContext(),"Values are:\n First value: "+value1+
        "\n Second Value: "+value2, Toast.LENGTH_LONG).show();
    }
    public void callFirstActivity(View view){
        Intent i = new Intent(getApplicationContext(), FirstActivity.class);
        startActivity(i);
    }
}

```

}



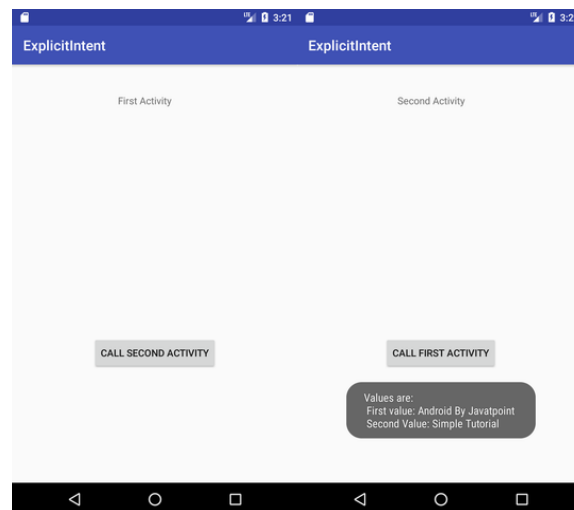
### Pre Lab VIVA Questions:

1. What is intent?
2. What is explicit intent?
3. State the difference between implicit and explicit intent.
4. What is Activity?
5. What are different life cycle phases of Activity?
6. What is the use of TextView component?
7. What are the different attributes of TextView Component?
8. What is the syntax of Button Component?
9. State the different attributes of Button Component and write Button component XML tag.
10. What does AndroidManifest.XML file contains?

### Post Lab VIVA Questions :

1. Write the Listener interface to be implemented for event handling on button click.
2. What is the class to be used for creating an intent?
3. explicit intent refers to calling an activity of other application or same application?
4. Write the code snippet of default activity onCreate().
5. What is method to be used to set the layout file to be displayed?

### Output:



## WEEK-2

**2a) Aim: Create an android app to perform mathematical operations(+,-,\*,/,%). (Use buttons, edittext, toast controls).**

### activity\_main.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="example.myapplication.com.sumoftwonumber.MainActivity">
```

#### <EditText

```
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="61dp"
    android:ems="10"
    android:inputType="number"
    tools:layout_editor_absoluteX="84dp"
    tools:layout_editor_absoluteY="53dp" />
```

#### <EditText

```
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editText1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="32dp"
    android:ems="10"
    android:inputType="number"
    tools:layout_editor_absoluteX="84dp"
    tools:layout_editor_absoluteY="127dp" />
```

#### <Button

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```

android:layout_below="@+id/editText2"
android:layout_centerHorizontal="true"
android:layout_marginTop="109dp"
android:text="ADD"
tools:layout_editor_absoluteX="148dp"
tools:layout_editor_absoluteY="266dp" />
</RelativeLayout>

```

### **MainActivity.java**

```

package example.myapplication.com.sumoftwonumber;
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.Toast;
public class MainActivity extends AppCompatActivity {
    private EditText edittext1, edittext2;
    private Button buttonSum;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButton();
    }
    public void addListenerOnButton() {
        edittext1 = (EditText) findViewById(R.id.editText1);
        edittext2 = (EditText) findViewById(R.id.editText2);
        buttonSum = (Button) findViewById(R.id.button);
        buttonSum.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String value1=edittext1.getText().toString();
                String value2=edittext2.getText().toString();
                int a=Integer.parseInt(value1);
                int b=Integer.parseInt(value2);
                int sum=a+b;
                Toast.makeText(getApplicationContext(),String.valueOf(sum),Toast.LENGTH_LONG).show();
            }
        });
    }
}

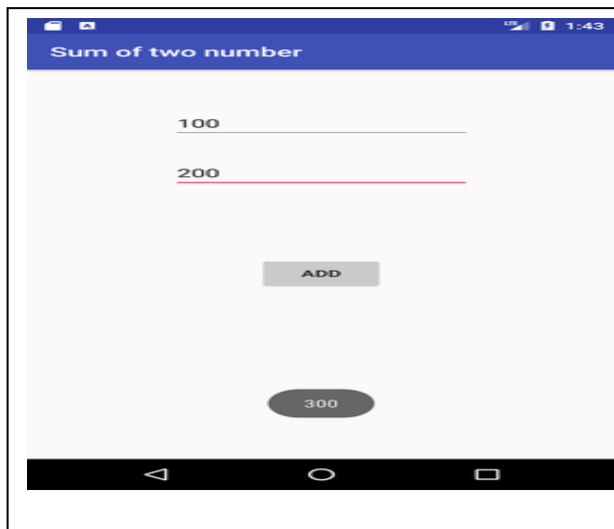
```

**Pre Lab VIVA Questions:**

1. What are different attributes of EditText Component?
2. What are the methods to be used for getting input from EditText and displaying some data on EditText component?
3. What are the classes and methods to be used for displaying some text as a message on a activity?
4. What is Activity?
5. What are different life cycle phases of Activity?
6. What is the use of TextView component?
7. What are the different attributes of TextView Component?
8. What is the syntax of Button Component?
9. State the different attributes of Button Component and write Button component XML tag.
10. What does AndroidManifest.XML file contains?

**Post Lab VIVA Questions :**

1. Write the Listener interface to be implemented for event handling on button click.
2. What is the use of setText()?
3. State the purpose of getText() and toString()?
4. Write the the different arguments to be supplied for makeText().
5. Method makeText() is available under which class?

**Output:**

**2b) Aim: Create an android app to display text in bold, italic, normal style with left, right, center alignments (use RadioButton, CheckBox controls)**

**activity\_main.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=".MainActivity">

    <CheckBox
        android:id="@+id/checkBox1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bold"
        android:textSize="25sp"/>

    <CheckBox
        android:id="@+id/checkBox2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Italic"
        android:textSize="25sp"
        android:layout_below="@+id/checkBox1"/>

    <CheckBox
        android:id="@+id/checkBox3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Normal"
        android:textSize="25sp"
        android:layout_below="@+id/checkBox2"/>

    <RadioGroup
        android:id="@+id/rg1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/checkBox3"
        <RadioButton
```

```
android:id="@+id/radioButton1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:checked="true"  
android:text="Left"  
android:textSize="25sp" />
```

```
<RadioButton  
    android:id="@+id/radioButton2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Right"  
    android:textSize="25sp"/>
```

```
<RadioButton  
    android:id="@+id/radioButton3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Center"  
    android:textSize="25sp" />  
</RadioGroup>
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginBottom="39dp"  
    android:text="Output!!"  
    android:textSize="35sp"  
    android:id="@+id/textView2" />
```

```
<TextView  
    android:id="@+id/t1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_above="@+id/textView2"  
    android:layout_alignParentEnd="true"  
    android:layout_marginBottom="17dp"  
    android:text=""  
    android:textSize="35sp" />
```

```
<TextView
```

```

android:id="@+id/t2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_above="@+id/t1"
android:layout_alignParentStart="true"
android:layout_marginBottom="14dp"
android:text=""
android:textSize="35sp" />

<TextView
android:id="@+id/t3"
android:layout_width="match_parent"
android:layout_height="wrap_content"

android:layout_alignParentStart="true"
android:layout_centerVertical="true"
android:text=""
android:textSize="35sp" />

<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentEnd="true"
android:layout_alignTop="@+id/checkbox3"
android:layout_marginEnd="46dp"
android:onClick="show"
android:text="Check"
android:textSize="35sp" />

</RelativeLayout>

```

### MainActivity.java

```

package com.example.administrator.week;

import android.graphics.Typeface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.RadioGroup;

```

```
import android.widget.TextView;
import org.w3c.dom.Text;
```

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void show(View view) {
        RadioGroup rg = (RadioGroup) findViewById(R.id.rg1);
        String value = ((RadioButton) findViewById(R.id.rg1)).getText().toString();
        CheckBox c1 = (CheckBox) findViewById(R.id.checkBox1);
        CheckBox c2 = (CheckBox) findViewById(R.id.checkBox2);
        CheckBox c3 = (CheckBox) findViewById(R.id.checkBox3);
        TextView t1 = (TextView) findViewById(R.id.t1);
        TextView t3 = (TextView) findViewById(R.id.t3);
        TextView t2 = (TextView) findViewById(R.id.t2);
        if(c1.isChecked()){
            t1.setText("BOLD");
            t1.setTypeface(Typeface.defaultFromStyle(Typeface.BOLD));
        }
        if(value.equals("Left"))
            t1.setGravity(Gravity.LEFT);
        else if(value.equals("Right"))
            t1.setGravity(Gravity.RIGHT);
        else t1.setGravity(Gravity.CENTER);
    }
    else t1.setText("");
    if(c2.isChecked()){
        t2.setText("ITALIC");
        t2.setTypeface(Typeface.defaultFromStyle(Typeface.ITALIC));
    }
    if(value.equals("Left"))
        t2.setGravity(Gravity.LEFT);
    else if(value.equals("Right"))
        t2.setGravity(Gravity.RIGHT);
    else t2.setGravity(Gravity.CENTER);
    }
    else t2.setText("");
    if(c3.isChecked()){
        t3.setText("NORMAL");
    }
    if(value.equals("Left"))
        t3.setGravity(Gravity.LEFT);
    else if(value.equals("Right"))
        t3.setGravity(Gravity.RIGHT);
    }
```



```
else t3.setGravity(Gravity.CENTER);  
    }  
else t3.setText("");  
    }  
}
```

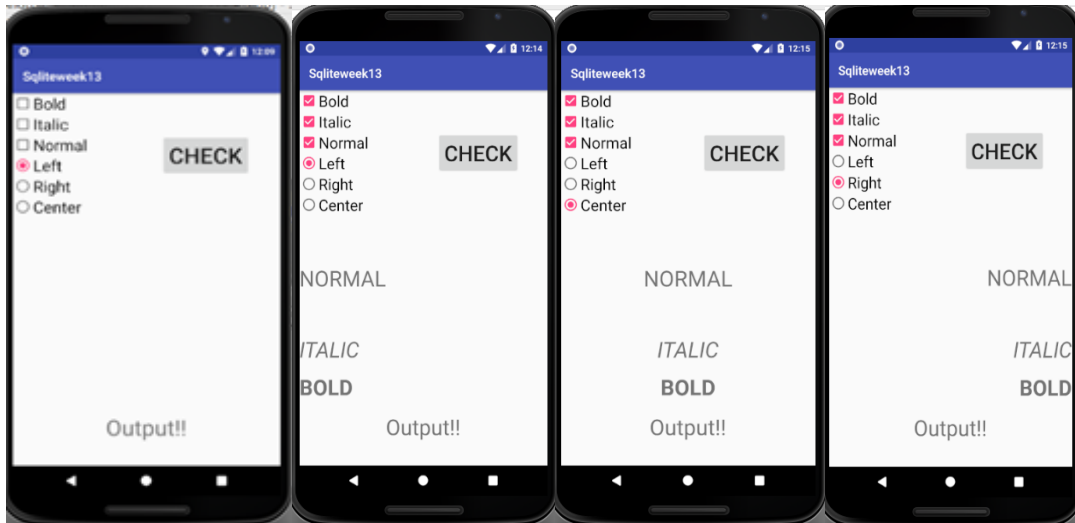
**Pre Lab VIVA Questions:**

1. What are different attributes of CheckBox, RadioButton and EditText Components?
2. What are the methods to be used for getting input from EditText and displaying some data on EditText component?
3. What are the classes and methods to be used for displaying some text as a message on a activity?
4. What is Activity?
5. What are different life cycle phases of Activity?
6. What is the use of TextView component?
7. What are the different attributes of TextView Component?
8. What is the syntax of Button Component?
9. State the different attributes of Button Component and write Button component XML tag.
10. What does AndroidManifest.XML file contains?

**Post Lab VIVA Questions :**

1. Write the Listener interface to be implemented for event handling on button click.
2. What is the method to be used for getting the selection status of CheckBox component?
3. What is the method to be used for fetching information of RadioButton Selected?
4. What is the use of setText()?
5. State the purpose of setGravity() and setTypeFace()?
6. Write the the different arguments to be supplied for makeText().
7. Method makeText() is available under which class?

**Output:**



## WEEK-3

3a) Aim: Create an android app to display name of the country from the list(Use spinner control)

### activity\_main.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="example.vardhaman.com.spinner.MainActivity">

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="149dp"
        android:layout_height="40dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.498" />

</android.support.constraint.ConstraintLayout>
```

### MainActivity.java

```
package example.vardhaman.com.spinner;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements
```

```

    AdapterView.OnItemClickListener {
String[] country = { "India", "USA", "China", "Japan", "Other"};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //Getting the instance of Spinner and applying OnItemSelectedListener on it
    Spinner spin = (Spinner) findViewById(R.id.spinner);
    spin.setOnItemSelectedListener(this);

    //Creating the ArrayAdapter instance having the country list
    ArrayAdapter aa = new ArrayAdapter(this,android.R.layout.simple_spinner_item,country);
    aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    //Setting the ArrayAdapter data on the Spinner
    spin.setAdapter(aa);
}
//Performing action onItemSelected and onNothing selected
@Override
public void onItemSelected(AdapterView<?> arg0, View arg1, int position, long id) {
    Toast.makeText(getApplicationContext(),country[position] , Toast.LENGTH_LONG).show();
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {
    // TODO Auto-generated method stub
}
}

```

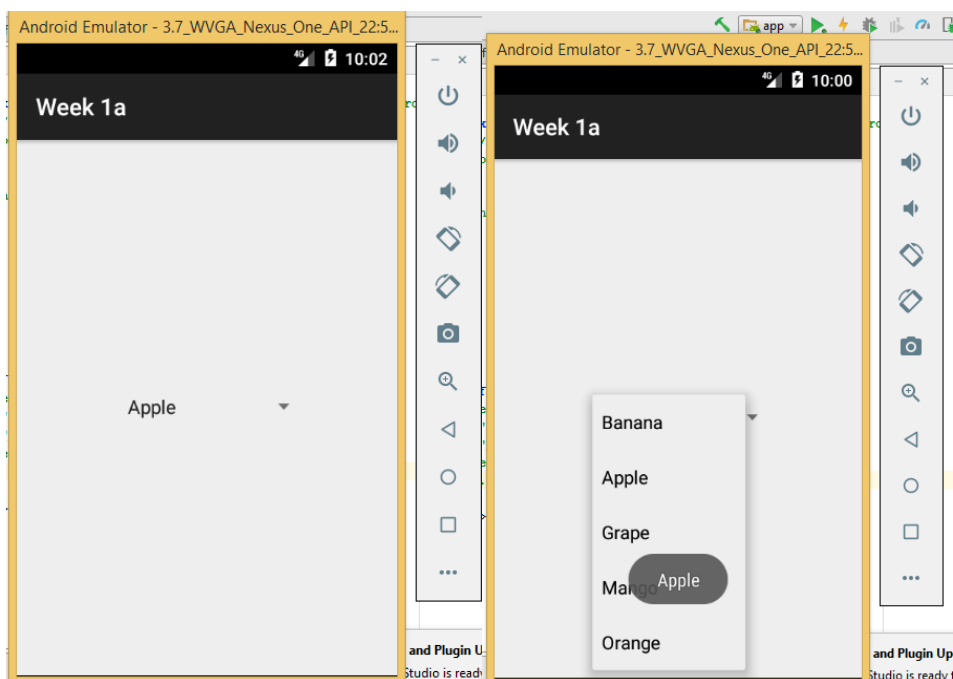
#### Pre Lab VIVA Questions:

1. What are different attributes of spinner component?
2. What is the purpose of spinner component?
3. What is array adapter?
4. What are the methods to be used for getting input from EditText and displaying some data on EditText component?
5. What are the classes and methods to be used for displaying some text as a message on a activity?
6. What is Activity?
7. What is the use of TextView component?
8. What are the different attributes of TextView Component?

### Post Lab VIVA Questions :

1. Write the Listener interface to be implemented for event handling on spinner component?.
2. What is the use of setAdapter()?.
3. What is the use of setText()?.
4. State the overridden methods to be implemented for performing action on selecting a item from a spinner component?
5. Write the the different arguments to be supplied for makeText().
6. Method makeText() is available under which class?

### Output:



### 3b) Aim: Create an android app to calculate age of a person (Use DatePicker control)

#### activity\_main.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.administrator.datepicker.MainActivity">

    <TextView
        android:id="@+id/textview1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/button1"
        android:layout_alignParentTop="true"
        android:text="Pick your DOB"
        android:textSize="30sp"
        android:layout_above="@+id/dp"/>

    <DatePicker
        android:id="@+id/dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/button1"
        android:layout_centerHorizontal="true">
</DatePicker>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="Calculate AGE" />
</RelativeLayout>
```

#### MainActivity.java

```
package com.example.administrator.datepicker;
```

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;
import android.util.StringBuilderPrinter;
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

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

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        final DatePicker picker1;
```

```
        Button displayDate;
```

```
        final TextView textView1;
```

```
        textView1=(TextView)findViewById(R.id.textView1);
```

```
        picker1=(DatePicker)findViewById(R.id.dp);
```

```
        displayDate=(Button)findViewById(R.id.button1);
```

```
        final int i=picker1.getYear();
```

```
        textView1.setText("pick your dob:");
```

```
        displayDate.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View view) {
```

```
                int j=picker1.getYear();
```

```
                textView1.setText(String.valueOf(i-j)+"years");
```

```
            }
```

```
        });
```

```
    }
```

```
}
```

#### Pre Lab VIVA Questions:

1. What are different attributes of Button, DatePicker and EditText Components?
2. What are the methods to be used for getting current year, month and day values?
3. What are the classes and methods to be used for displaying some text as a message on a activity?
4. What is Activity?
5. What are different life cycle phases of Activity?
6. What is the use of TextView component?
7. What are the different attributes of TextView Component?
8. What is the XML syntax of Button Component?

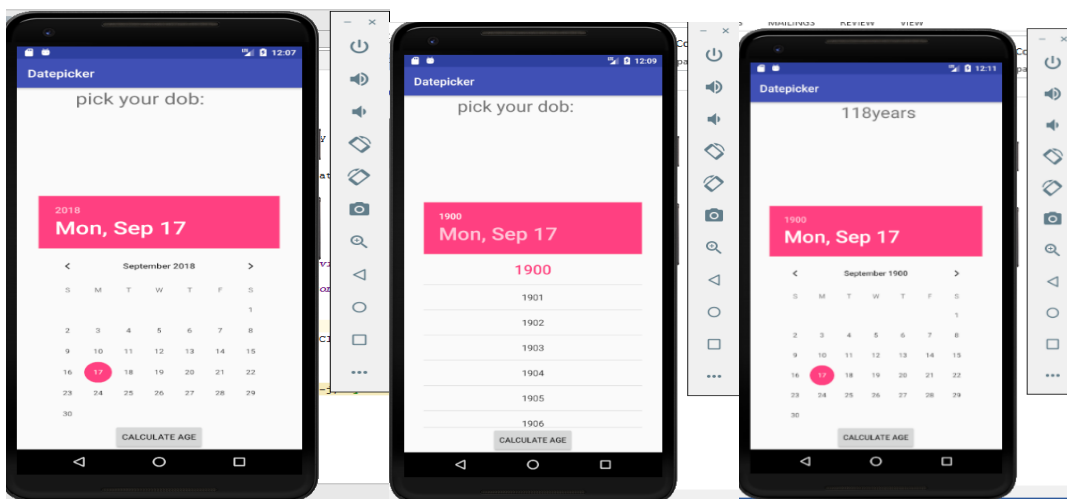


9. What does AndroidManifest.XML file contains?

### Post Lab VIVA Questions :

1. Write the Listener interface to be implemented for event handling on button click.
2. What is the use of `getYear()`?
3. What is the use of `setText()`?
4. Write the the different arguments to be supplied for `makeText()`.
5. Method `makeText()` is available under which class?

### OutPut:



## WEEK-4

### 4) Aim: Create an android app for login control and validate login details .

#### activity\_main.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="e.administrator.week1a.MainActivity">

        <EditText
            android:id="@+id/editText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="8dp"
            android:layout_marginStart="8dp"
            android:layout_marginTop="136dp"
            android:ems="10"
            android:hint="user ID"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.575"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <EditText
            android:id="@+id/editText2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="8dp"
            android:layout_marginStart="8dp"
            android:layout_marginTop="204dp"
            android:ems="10"
            android:hint="Password"
            android:inputType="textPassword"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.575"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="120dp"
    android:text="Visit"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />
</android.support.constraint.ConstraintLayout>
```

### MainActivity.java

```
package e.administrator.week1a;
```

```
import android.content.Intent;
import android.net.Uri;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends Activity {
```

```
    Button button;
    EditText editText,editText1;
    String w="vardhaman";
    String q="cse";
```

```
@Override
```

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

```
    button = findViewById(R.id.button);
    editText = findViewById(R.id.editText);
    editText1 = findViewById(R.id.editText2);
```

```

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String url = editText.getText().toString();
        String pwd = editText1.getText().toString();
        if (w.equals(url)&&q.equals(pwd)) {
            Toast.makeText(getApplicationContext(), "success" + url, Toast.LENGTH_LONG).show();
            Intent i=new Intent(getApplicationContext(),HomeActivity.class);
            startActivity(i);
        }
        else{
            Toast.makeText(getApplicationContext(), "! Authentication Failed" + url,
            Toast.LENGTH_LONG).show();
        }
    }
});
}
}

```

### Android Manifest.xml

```

<activity
    android:name=".HomeActivity"
    android:label="@string/title_activity_home"></activity>
</application>

```

### activity\_home.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="e.administrator.week1a.HomeActivity"
    android:background="#00ff00">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="WELCOME TO HOME"/>
</android.support.constraint.ConstraintLayout>

```

## HomeActivity.java

```
package e.administrator.week1a;

import android.os.Bundle;
import android.app.Activity;

public class HomeActivity extends Activity {

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

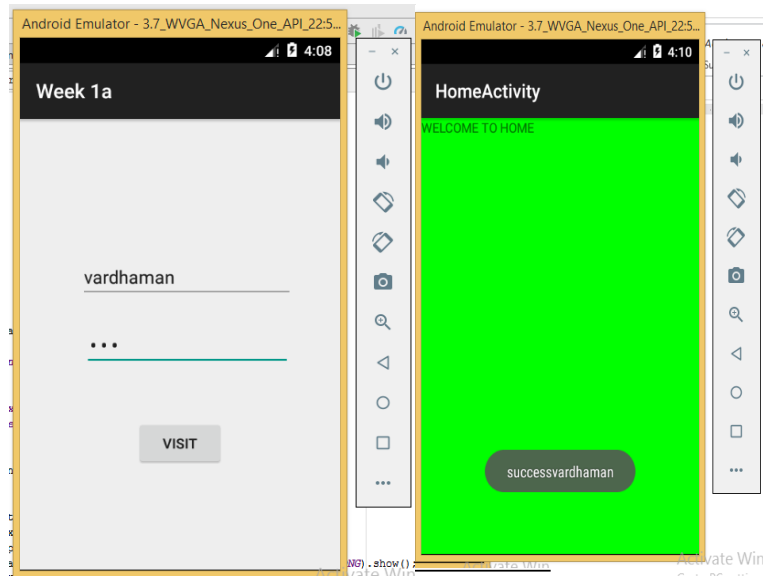
### **Pre Lab VIVA Questions:**

1. What are different attributes of Button and EditText Components?
2. What are the methods to be used for getting input from EditText and displaying some data on EditText component?
3. What are the classes and methods to be used for displaying some text as a message on a activity?
4. What is explicit intent?
5. What is Activity?
6. What is the use of TextView component?
7. What are the different attributes of TextView Component?
8. What does AndroidManifest.XML file contains?

### **Post Lab VIVA Questions :**

1. Write the Listener interface to be implemented for event handling on button click.
2. What is the method used for validating username and password entered?
3. What is the use of setText()?
4. In this program, implicit or explicit intent is called on clicking a button?
5. Write the the different arguments to be supplied for makeText().
6. Method makeText() is available under which class?

## OutPut:



## WEEK-5

5a) Aim: Create an android app to demonstrate AlertDialog.

### activity\_main.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="example.vardhaman.com.alertdialog.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:text="Close app"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

### strings.xml

```
<resources>
<string name="app_name">AlertDialog</string>
<string name="dialog_message">Welcome </string>
<string name="dialog_title">vardhaman</string>
</resources>
```

### MainActivity.java

```
package example.vardhaman.com.alertdialog;

import android.content.DialogInterface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
```



```

import android.widget.Button;
import android.app.AlertDialog;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button closeButton;
    AlertDialog.Builder builder;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        closeButton = (Button) findViewById(R.id.button);
        builder = new AlertDialog.Builder(this);
        closeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                //Uncomment the below code to Set the message and title from the strings.xml file
                builder.setMessage(R.string.dialog_message) .setTitle(R.string.dialog_title);

                //Setting message manually and performing action on button click
                builder.setMessage("Do you want to close this application ?")
                    .setCancelable(false)
                    .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
                            finish();
                            Toast.makeText(getApplicationContext(),"you choose yes action for alertbox",
                                Toast.LENGTH_SHORT).show();
                        }
                    })
                    .setNegativeButton("No", new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int id) {
                            // Action for 'NO' Button
                            dialog.cancel();
                            Toast.makeText(getApplicationContext(),"you choose no action for alertbox",
                                Toast.LENGTH_SHORT).show();
                        }
                    });
                //Creating dialog box
                AlertDialog alert = builder.create();
                //Setting the title manually
                alert.setTitle("AlertDialogExample");
                alert.show();
            }
        });
    }
}

```

```
}  
});  
}  
}
```

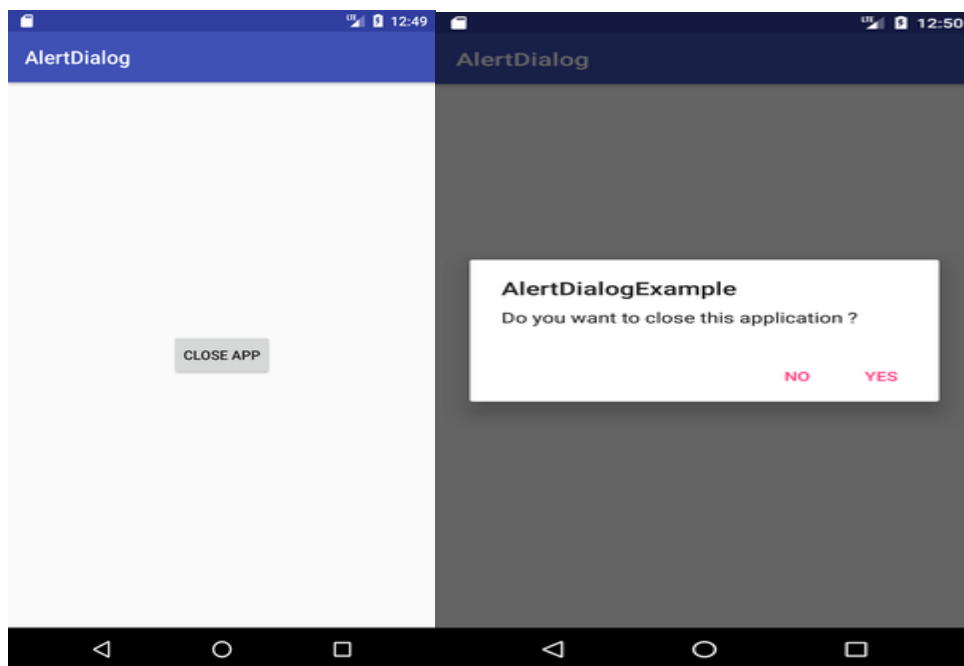
### Pre Lab VIVA Questions:

1. What are different attributes of Button Component?
2. What are the different components of AlertDialog?
3. What is Activity?
4. What is DialogInterface?
5. State the purpose of onClick().

### Post Lab VIVA Questions :

1. Write the Listener interface to be implemented for event handling on button click.
2. What is the use of the methods builderobject.create() and setTitle()?
3. What are the methods to be written for setting action buttons to AlertDialog component?
4. Write the code snippet for setting postive and negative buttons with action.

### Output:



## 5b) Aim: Create an android app to demonstrate WebView control

### activity\_main.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="example.vardhaman.com.webview.MainActivity">

    <WebView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/webView"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

### MainActivity.java

```
package example.vardhaman.com.webview;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        WebView mywebview = (WebView) findViewById(R.id.webView);
        // mywebview.loadUrl("https://www.vardhaman.org/");

        String data = "<html><body><h1>WELCOME TO VARDHAMAN CSE!!!h1></body></html>";
        mywebview.loadData(data, "text/html", "UTF-8");
    }
}
```

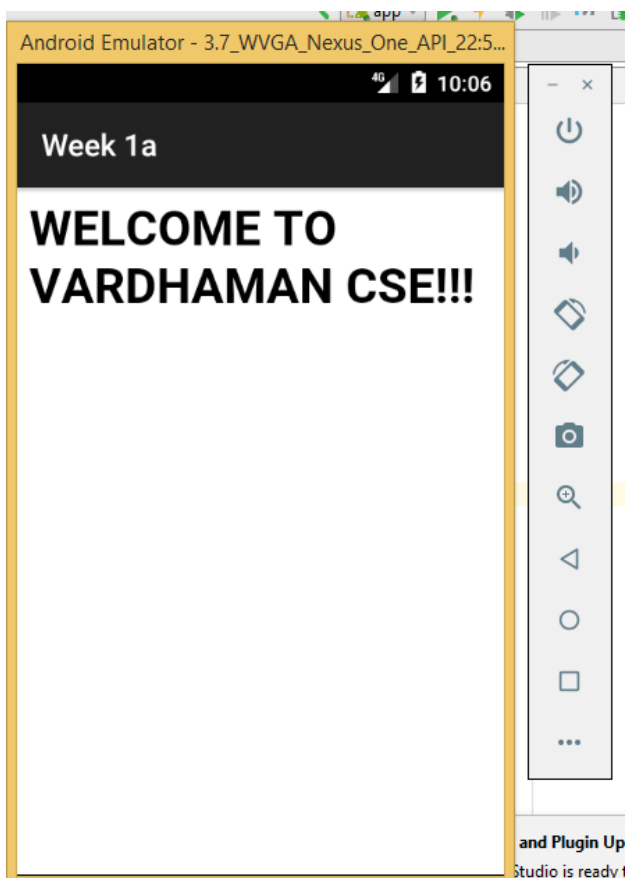
}  
}

**Pre Lab VIVA Questions:**

1. What is the class to be used for adding WebView component.
2. What is the purpose of WebView Component.
3. What is Activity?
4. What are the different attributes of WebView Component?
5. What does AndroidManifest.XML file contains?

**Post Lab VIVA Questions :**

1. What is the use of loadData()?
2. State the purpose of loadUrl()?
3. State the difference between loadData() and loadUrl()?
4. How to display the HTML data on a activity using WebView component?

**Output:**



## WEEK-6

6a) Aim: Create an android app to show Analog and Digital clocks

### activity\_main.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="example.vardhaman.com.analogdigital.MainActivity">

    <AnalogClock
        android:id="@+id/analogClock1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="136dp"
        android:layout_marginTop="296dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <DigitalClock
        android:id="@+id/digitalClock1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/analogClock1"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="176dp"
        android:layout_marginTop="84dp"
        android:text="DigitalClock"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```



### **MainActivity.java**

```
package example.vardhaman.com.analogdigital;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

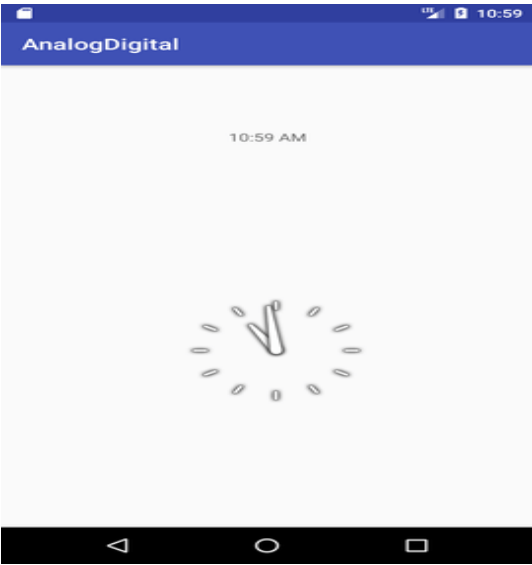
#### **Pre Lab VIVA Questions:**

1. What are different attributes of analogclock and DigitalClock components?
2. What are names of AnalogClock and DigitalClock components in the current version of Android SDK?
3. What is Activity?
4. What does AndroidManifest.XML file contains?

#### **Post Lab VIVA Questions :**

1. What are the parent classes of Analog clock and DigitalClock components?
2. What is TextClock Component?
3. What are the different attributes of TextClock Component?

#### **Output:**



**6b) Aim: Create an android app to illustrate a progress bar.**

**activity\_main.xml**

```
<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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">
    <ProgressBar
        android:id="@+id/simpleProgressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="invisible"
        android:layout_centerHorizontal="true"/>
    <Button
        android:id="@+id/startButton"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="Start"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:padding="10dp"
        android:background="#0f0"
        android:textColor="#fff"/>
</RelativeLayout>
```

**MainActivity.java**

```
package example.gb.progressbarexample;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
```

```

import android.view.View;
import android.widget.ProgressBar;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // initiate progress bar and start button
        final ProgressBar simpleProgressBar = (ProgressBar) findViewById(R.id.simpleProgressBar);
        Button startButton = (Button) findViewById(R.id.startButton);
        // perform click event on button
        startButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // visible the progress bar
                simpleProgressBar.setVisibility(View.VISIBLE);
            }
        });
    }
}

```

#### **Pre Lab VIVA Questions:**

1. What are different attributes of ProgressBar and Button Components?
2. What does ProgressBar component represents?
3. What are the 2 different styles of ProgressBar?
4. What is Activity?
5. State the purpose of setOnClickListener().

#### **Post Lab VIVA Questions :**

1. Write the Listener interface to be implemented for event handling on button click.
2. Write the code snippet for Listener to be implemented for event handling on ProgressBar component.
3. What is the default visibility status of ProgressBar component?
4. State the use of setVisibility().

**Output:**



## WEEK-7

### 7) Aim: Create an android app to demonstrate option menu with handling listeners

#### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout 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="example.myapplication.com.optionmenu.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

</android.support.design.widget.CoordinatorLayout>
```

#### context\_main.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">
```

```
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="example.myapplication.com.optionmenu.MainActivity"
tools:showIn="@layout/activity_main">
```

### **<TextView**

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

**</android.support.constraint.ConstraintLayout>**

### **menu\_main.xml**

```
<menu 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"
    tools:context="example.myapplication.com.optionmenu.MainActivity">

    <item android:id="@+id/item1"
        android:title="Item 1"/>
    <item android:id="@+id/item2"
        android:title="Item 2"/>
    <item android:id="@+id/item3"
        android:title="Item 3"
        app:showAsAction="withText"/>
</menu>
```

### **MainActivity.java**

```
package example.myapplication.com.optionmenu;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;
```

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        switch (id){
            case R.id.item1:
                Toast.makeText(getApplicationContext(),"Item 1 Selected",Toast.LENGTH_LONG).show();
                return true;
            case R.id.item2:
                Toast.makeText(getApplicationContext(),"Item 2 Selected",Toast.LENGTH_LONG).show();
                return true;
            case R.id.item3:
                Toast.makeText(getApplicationContext(),"Item 3 Selected",Toast.LENGTH_LONG).show();
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
}

```

#### Pre Lab VIVA Questions:

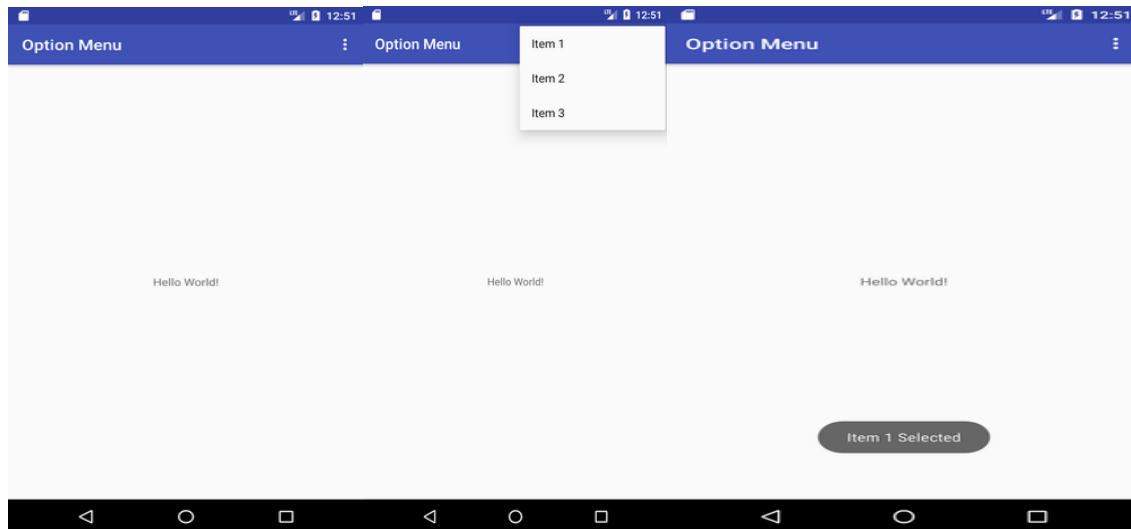
- 1.What are attributes of ToolBar component?
- 2.What is the use of toolbar component?
- 3.What are the use of inflate method?
- 4.What are the constants of Toast class?
- 5.How to add items in the menu?



### Post Lab VIVA Questions :

1. Write the Listener interface to be implemented for event handling on selection of an item.
2. What is the method to retrieve the item id of the selected item.
3. Write a sample code to display the selected item.
4. What is the use of `setSupportActionBar` method?
5. State the different parameters in the `makeText` method of `Toast`.

### Output:



## WEEK-8

**8) Aim: Create an android app to scroll list of images and display details of images (name, size etc) using ImageSwitcher control.**

### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout 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"
    android:fitsSystemWindows="true"
    tools:context="com.example.test.imageswitcher.MainActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/content_main" />

</android.support.design.widget.CoordinatorLayout>
```

### content\_main.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">
```

```
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="com.example.test.imageswitcher.MainActivity"
tools:showIn="@layout/activity_main">
```

#### **<TextView**

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Image Switcher Example"
```

```
android:id="@+id/textView"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true" />
```

#### **<ImageSwitcher**

```
android:id="@+id/imageSwitcher"
android:layout_width="match_parent"
android:layout_height="250dp"
android:layout_marginBottom="28dp"
android:layout_marginTop="40dp" />
```

#### **<Button**

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Next"
android:id="@+id/button"
android:layout_marginBottom="47dp"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true" />
```

**</RelativeLayout>**

#### **MainActivity.java**

```
package com.example.test.imageswitcher;
```

```
import android.os.Bundle;
```

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

```
import android.support.v7.widget.Toolbar;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.ImageSwitcher;
import android.widget.ImageView;
import android.widget.ViewSwitcher;
```

```
import android.app.ActionBar;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
```

```
public class MainActivity extends AppCompatActivity {
    ImageSwitcher imageSwitcher;
    Button nextButton;

    int imageSwitcherImages[] = {R.drawable.cpp, R.drawable.c_sarp, R.drawable.jsp, R.drawable.mysq
        l, R.drawable.hadoop};

    int switcherImageLength = imageSwitcherImages.length;
    int counter = -1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        imageSwitcher = (ImageSwitcher) findViewById(R.id.imageSwitcher);
        nextButton = (Button) findViewById(R.id.button);

        imageSwitcher.setFactory(new ViewSwitcher.ViewFactory() {
            @Override
            public View makeView() {
                ImageView switcherImageView = new ImageView(getApplicationContext());
                switcherImageView.setLayoutParams(new ImageSwitcher.LayoutParams(
                    ActionBar.LayoutParams.FILL_PARENT, ActionBar.LayoutParams.FILL_PARENT ));
                switcherImageView.setScaleType(ImageView.ScaleType.FIT_CENTER);
                switcherImageView.setImageResource(R.drawable.hadoop);
                //switcherImageView.setMaxHeight(100);
                return switcherImageView;
            }
        });

        Animation aniOut = AnimationUtils.loadAnimation(this, android.R.anim.slide_out_right);
        Animation aniIn = AnimationUtils.loadAnimation(this, android.R.anim.slide_in_left);
```

```

imageSwitcher.setOutAnimation(aniOut);
imageSwitcher.setInAnimation(aniIn);

nextButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        counter++;
        if (counter == switcherImageLength){
            counter = 0;
            imageSwitcher.setImageResource(imageSwitcherImages[counter]);
        }
        else{
            imageSwitcher.setImageResource(imageSwitcherImages[counter]);
        }
    }
});
}

}

```

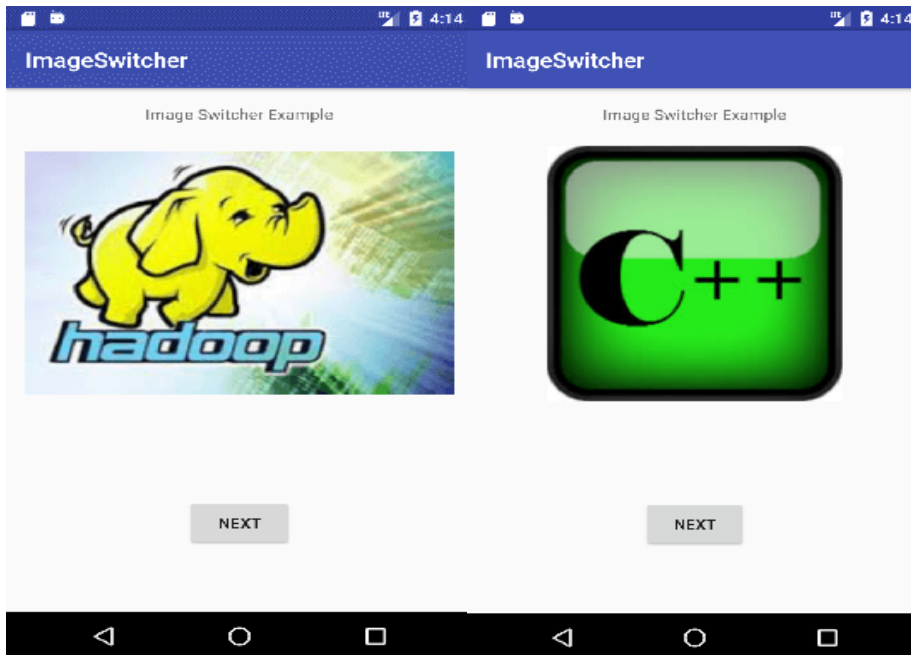
#### **Pre Lab VIVA Questions:**

1. What is the use of ImageSwitcher component?
2. State the usage of ImageView.
3. What is the use of Animation component?
4. What are the different methods used in Animation?
5. What is the method used to set maximum height of the image?
6. Write about ToolBar component.
7. What does AndroidManifest.XML file contains?
8. State different methods under ImageSwitcher component.
9. What are the parameters of setImageResource method?

#### **Post Lab VIVA Questions .:**

1. Write the code snippet of default activity onCreate().
2. State the usage and different parameters used in setLayoutParams method .
3. What is the method used to set image in the ImageView?
4. Write the methods to apply sliding effects using Animation.
5. What is the interface implemented under setfactory().
6. Write the Listener interface to be implemented for event handling on clicking a button.

#### **Output:**



## WEEK 9

### 9a) Aim: Create an android app to demonstrate mediaplayer

#### MainActivity.java

```
package com.example.audioplay;
```

```
import android.media.MediaPlayer;
```

```
import android.os.Bundle;
```

```
import android.os.Environment;
```

```
import android.app.Activity;
```

```
import android.view.Menu;
```

```
import android.view.View;
```

```
import android.view.View.OnClickListener;
```

```
import android.widget.Button;
```

```
public class MainActivity extends Activity {
```

```
    Button start,pause,stop;
```

```
    @Override
```

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

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        start=(Button)findViewById(R.id.button1);
```

```
        pause=(Button)findViewById(R.id.button2);
```

```
        stop=(Button)findViewById(R.id.button3);
```

```
        //creating media player
```

```
        final MediaPlayer mp=new MediaPlayer();
```

```
        try{
```

```
            //you can change the path, here path is external directory(e.g. sdcard) /Music/main.mp3
```

```
            mp.setDataSource(Environment.getExternalStorageDirectory().getPath()+"/Music/main.mp3");
```

```
            mp.prepare();
```

```
        }catch(Exception e){e.printStackTrace();}
```

```
        start.setOnClickListener(new OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                mp.start();
```

```

    }
});
pause.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        mp.pause();
    }
});
stop.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        mp.stop();
    }
});
}
}

```

### activity\_main.xml

```

<RelativeLayout xmlns:androclass="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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="30dp"
        android:text="Audio Controller" />

    <Button
        android:id="@+id/button1"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"

```



```
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView1"
android:layout_below="@+id/textView1"
android:layout_marginTop="48dp"
android:text="start" />
```

```
<Button
    android:id="@+id/button2"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button1"
    android:layout_toRightOf="@+id/button1"
    android:text="pause" />
```

```
<Button
    android:id="@+id/button3"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/button2"
    android:layout_toRightOf="@+id/button2"
    android:text="stop" />
```

```
</RelativeLayout>
```

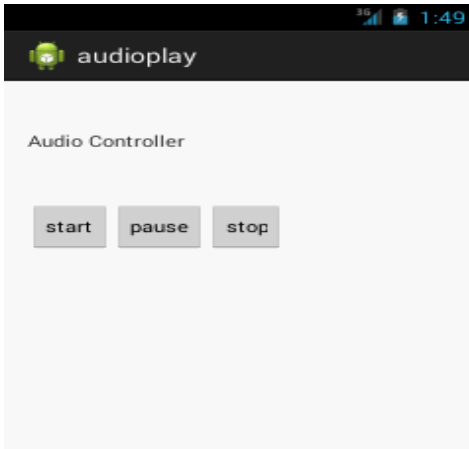
#### **Pre Lab VIVA Questions:**

- 1.What are the methods of media player class?
- 2.What is the use of method setdatasource()?
- 3.What is the usage of start(), stop(), pause()?
- 4.Which method is used for synchronous playback?
- 5.What is the syntax of setDataSource()?
6. What is the syntax of Button Component?
- 7.What does AndroidManifest.XML file contains?

#### **Post Lab VIVA Questions :**

- 1.What is the use of MediaPlayer class?
- 2Which method is used to check whether the MediaPlayer is working or not?
- 3.What are the components used to implement mediaplayer?
- 4.What is the use of method getExternalStorageDirectory?
- 5.What are the parameters of setVolume()?

#### **Output:**



9b) Aim: Create an android app to show details of phone contacts and implement calling, receiving

**activity\_main.xml**

```
<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"
    tools:context=".MainActivity" >
```

**<Button**

```
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="118dp"
    android:text="Call" />
```

**<EditText**

```
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="25dp"
    android:ems="10" />
```

```
</RelativeLayout>
```

## Android Manifest.xml

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

## MainActivity.java

```
package com.example.phonecall;
```

```
import android.net.Uri;
```

```
import android.os.Bundle;
```

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```
import android.view.Menu;
```

```
import android.view.View;
```

```
import android.view.View.OnClickListener;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
public class MainActivity extends Activity {
```

```
    EditText edittext1;
```

```
    Button button1;
```

```
    @Override
```

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

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        //Getting the edittext and button instance
```

```
        edittext1=(EditText)findViewById(R.id.editText1);
```

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

```
        //Performing action on button click
```

```
        button1.setOnClickListener(new OnClickListener(){
```

```
            @Override
```

```
            public void onClick(View arg0) {
```

```
                String number=edittext1.getText().toString();
```

```
                Intent callIntent = new Intent(Intent.ACTION_CALL);
```

```
                callIntent.setData(Uri.parse("tel:"+number));
```

```
                startActivity(callIntent);
```

```
            }
```

```
        });
```

```
}
```

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.activity_main, menu);  
    return true;  
} }
```

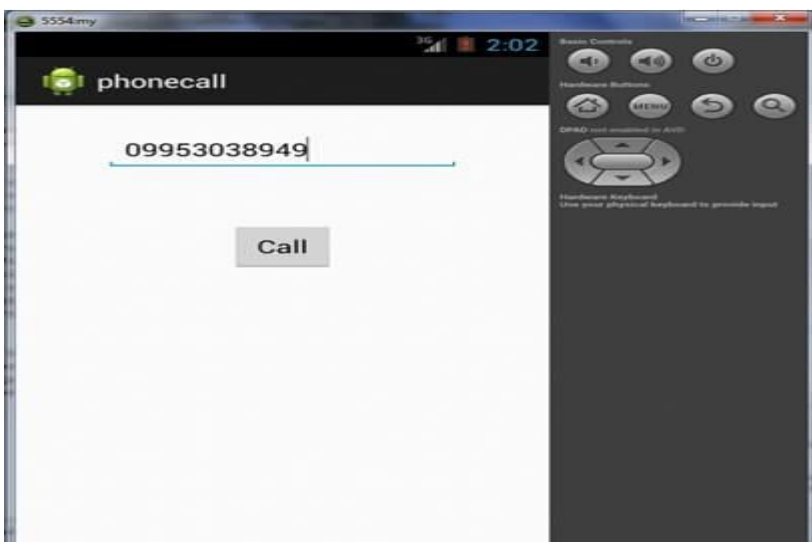
### Pre Lab VIVA Questions:

- 1.What is Intent?
- 2.What is the use of method parse()?
- 3.What is the use of method getMenuInflater()?
- 4.What is the constant used in intent for making a call?
- 5.What are the package used for onClickListener interface?
- 6.What does AndroidManifest.XML file contains?

### Post Lab VIVA Questions :

- 1.Write the syntax of intent?
- 2.How to implement onClickListener?
- 3.What is the syntax of inflate()?
- 4.What is the role of startActivity()?
- 5.What permissions should be added in AndroidManifest.xml to access phone call application?

### Output:



## WEEK 10 (Camera Application)

### 10) Aim: Create an android app to demonstrate camera.

By the help of 2 constants of **MediaStore** class, we can capture picture and video without using the instance of Camera class.

1. ACTION\_IMAGE\_CAPTURE
2. ACTION\_VIDEO\_CAPTURE

#### activity\_main.xml

```
<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"
    tools:context=".MainActivity" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="Take a Photo" >
    </Button>
    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_above="@+id/button1"
        android:layout_alignParentTop="true"
        android:src="@drawable/ic_launcher" >
    </ImageView>
</RelativeLayout>
```

## MainActivity.java

```
package com.example.camera;
```

```
import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
```

```
public class MainActivity extends Activity {
    private static final int CAMERA_REQUEST = 1888;
    ImageView imageView;
    public void onCreate(Bundle savedInstanceState) {

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

        imageView = (ImageView) this.findViewById(R.id.imageView1);
        Button photoButton = (Button) this.findViewById(R.id.button1);

        photoButton.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cameraIntent, CAMERA_REQUEST);
            }
        });
    }

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        if (requestCode == CAMERA_REQUEST) {
            Bitmap photo = (Bitmap) data.getExtras().get("data");
            imageView.setImageBitmap(photo);
        }
    }
}
```

```
}
```

```
@Override
```

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.activity_main, menu);  
    return true;  
}
```

```
}
```

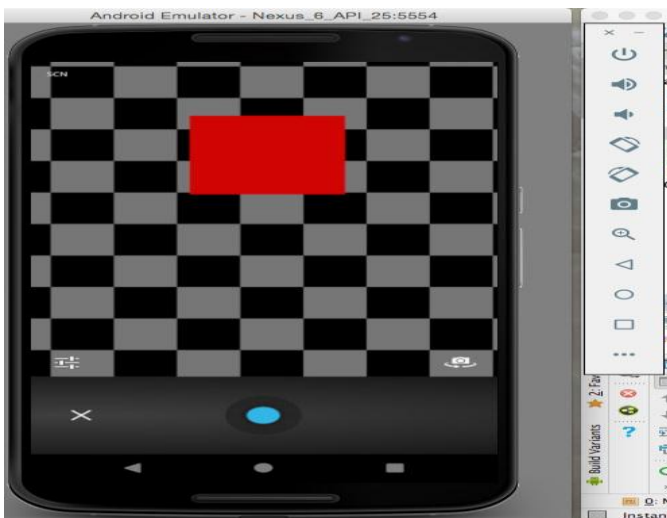
### Pre Lab VIVA Questions:

- 1.What are the constants of MediaStore class?
- 2.What are the components of Camera?
- 3.What is the use of ACTION\_IMAGE\_CAPTURE?
- 4.What is the use of getExtras()?
- 5.What does setImageBitmap() do?
6. State the different attributes of Button Component and write Button component XML tag.
- 7.What does AndroidManifest.XML file contains?

### Post Lab VIVA Questions :

- 1.What are the parameters passed in onActivityResult()?
- 2.What is the syntax of inflate()?
- 3.What is the constant used for capturing video?
- 4.What is the use of MediaRecorder?
- 5..What is the difference between MediaStore and MediaPlayer class?
6. What is method to be used to set the layout file to be displayed?

### Output:



## WEEK 11

### 11a) Aim: Create an android app to demonstrate sending e-mail

#### activity\_main.xml

```
<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"
    tools:context=".MainActivity" >
```

#### <EditText

```
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="22dp"
    android:layout_marginTop="16dp"
    android:ems="10" />
```

#### <EditText

```
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editText1"
    android:layout_below="@+id/editText1"
    android:layout_marginTop="18dp"
    android:ems="10" >
```

```
<requestFocus />
```

```
</EditText>
```

#### <EditText

```
    android:id="@+id/editText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editText2"
    android:layout_below="@+id/editText2"
    android:layout_marginTop="28dp"
```



```
android:ems="10"  
android:inputType="textMultiLine" />
```

#### **<TextView**

```
android:id="@+id/textView1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBaseline="@+id/editText1"  
android:layout_alignBottom="@+id/editText1"  
android:layout_alignParentLeft="true"  
android:text="To:" />
```

#### **<TextView**

```
android:id="@+id/textView2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBaseline="@+id/editText2"  
android:layout_alignBottom="@+id/editText2"  
android:layout_alignParentLeft="true"  
android:text="Subject:" />
```

#### **<TextView**

```
android:id="@+id/textView3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignBaseline="@+id/editText3"  
android:layout_alignBottom="@+id/editText3"  
android:layout_alignParentLeft="true"  
android:text="Message:" />
```

#### **<Button**

```
android:id="@+id/button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/editText3"  
android:layout_below="@+id/editText3"  
android:layout_marginLeft="76dp"  
android:layout_marginTop="20dp"  
android:text="Compose an email" />
```

</RelativeLayout>

## MainActivity.java

```
package com.example.sendemail;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity {
    EditText editTextTo,editTextSubject,editTextMessage;
    Button send;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editTextTo=(EditText)findViewById(R.id.editText1);
        editTextSubject=(EditText)findViewById(R.id.editText2);
        editTextMessage=(EditText)findViewById(R.id.editText3);

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

        send.setOnClickListener(new OnClickListener(){

            @Override
            public void onClick(View arg0) {
                String to=editTextTo.getText().toString();
                String subject=editTextSubject.getText().toString();
                String message=editTextMessage.getText().toString();

                Intent email = new Intent(Intent.ACTION_SEND);
                email.putExtra(Intent.EXTRA_EMAIL, new String[]{ to});
                email.putExtra(Intent.EXTRA_SUBJECT, subject);
                email.putExtra(Intent.EXTRA_TEXT, message);
```

```

//need this to prompts email client only
email.setType("message/rfc822");

startActivity(Intent.createChooser(email, "Choose an Email client :"));

}

});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.activity_main, menu);
return true;
}

}

```

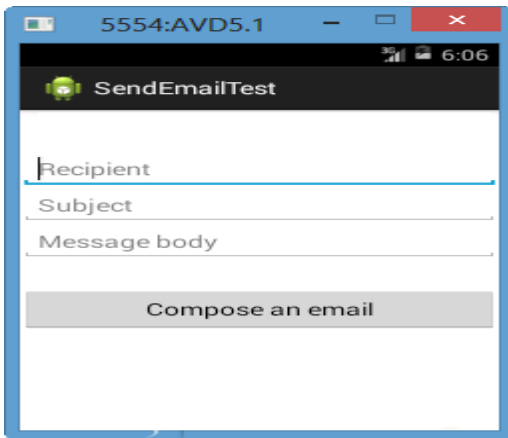
#### Pre Lab VIVA Questions:

1. What is constant used for sending an e-mail?
2. Write sample code to send email?
3. What is the type of email i.e. used in setType()?
4. What attributes are present in putextras()?
5. Which method is used to add items to the ActionBar?
6. State the different attributes of Button Component and write Button component XML tag.
7. What does AndroidManifest.XML file contains?

#### Post Lab VIVA Questions :

1. Write syntax of startActivity() in sending an email.
2. What is method to be used to set the layout file to be displayed?
3. How to convert objects into string?
4. What type of intent is used to send an email?
5. What is the use of createChooser() in intent?
6. What are the parameters of createChooser()?

**Output:**



## 11b) Aim: Create an android app to demonstrate sending SMS

### activity\_main.xml

```
<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"
    tools:context=".MainActivity" >
```

#### <EditText

```
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="20dp"
    android:ems="10" />
```

#### <EditText

```
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editText1"
    android:layout_below="@+id/editText1"
    android:layout_marginTop="26dp"
    android:ems="10"
    android:inputType="textMultiLine" />
```

#### <TextView

```
    android:id="@+id/textView1"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/editText1"
android:layout_alignBottom="@+id/editText1"
android:layout_toLeftOf="@+id/editText1"
android:text="Mobile No:" />
```

#### **<TextView**

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/editText2"
android:layout_alignBottom="@+id/editText2"
android:layout_alignLeft="@+id/textView1"
android:text="Message:" />
```

#### **<Button**

```
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText2"
android:layout_below="@+id/editText2"
android:layout_marginLeft="34dp"
android:layout_marginTop="48dp"
android:text="Send SMS" />
```

**</RelativeLayout>**

#### **Android-Manifest.xml**

```
<uses-permission android:name="android.permission.SEND_SMS"/>
```

#### **MainActivity.java**

```
package com.example.sendsms;

import android.os.Bundle;
import android.app.Activity;
import android.app.PendingIntent;
import android.content.Intent;
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.View;
```

```

import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {

    EditText mobileno,message;
    Button sendsms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mobileno=(EditText)findViewById(R.id.editText1);
        message=(EditText)findViewById(R.id.editText2);
        sendsms=(Button)findViewById(R.id.button1);

        //Performing action on button click
        sendsms.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                String no=mobileno.getText().toString();
                String msg=message.getText().toString();

                //Getting intent and PendingIntent instance
                Intent intent=new Intent(getApplicationContext(),MainActivity.class);
                PendingIntent pi=PendingIntent.getActivity(getApplicationContext(), 0, intent,0);

                //Get the SmsManager instance and call the sendTextMessage method to send message
                SmsManager sms=SmsManager.getDefault();
                sms.sendTextMessage(no, null, msg, pi,null);

                Toast.makeText(getApplicationContext(), "Message Sent successfully!",
                Toast.LENGTH_LONG).show();
            } });
        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is present.
            getMenuInflater().inflate(R.menu.activity_main, menu);

```

```
return true;
}
}
```

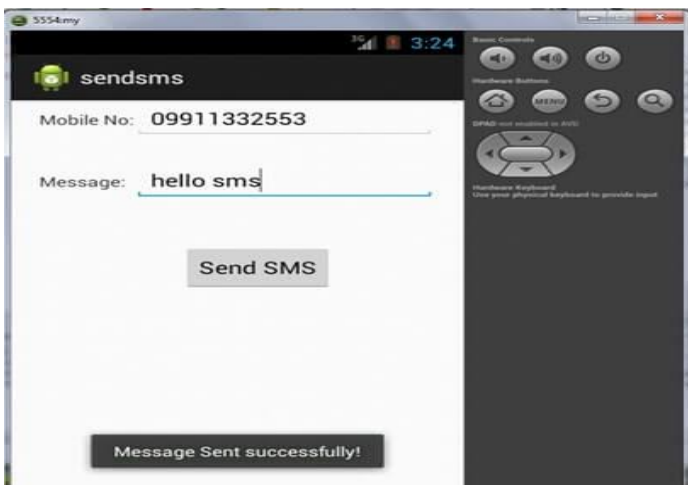
### Pre Lab VIVA Questions:

- 1.What is the use of PendingIntent?
- 2.Which type of Intent is used?
- 3.What are the classes used to send sms?
- 4.Explain the PendingIntent.getActivity(),SmsManager.getDefault(),sendTextMessage() methods?
- 5.What are the attributes of getActivity() in pendingIntent?
6. What is the use of TextView component?
7. What are the different attributes of TextView Component?
8. What is the syntax of Button Component?

### Post Lab VIVA Questions :

- 1.What permissions should be added in AndroidManifest.xml?
- 2.Write the syntax of Intent.
- 3.What is the method used to send message?
- 4.What is the use of smsManager?
- 5.What are the attributes of sendTextMessage()?
- 6.Write the syntax of PendingIntent.

### Output:





## WEEK 12

- 12) Aim:** Create an android app to store details of students in SQLite and display the details perform insert, update, delete operations on student database

### activity\_main.xml

```
<AbsoluteLayoutxmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="110dp"
    android:text="Enter Rollno:"
    android:textSize="20sp" />
```

```
<EditText
    android:id="@+id/Rollno"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="100dp"
    android:inputType="number"
    android:textSize="20sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="160dp"
    android:text="Enter Name:"
    android:textSize="20sp" />
```

```
<EditText
    android:id="@+id/Name"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="150dp"
    android:inputType="text"
    android:textSize="20sp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
    android:textSize="20sp" />
```

```
<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />
```

```
<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />
```

```
<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />
```

```
<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />
```

```
<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="400dp"
    android:text="View"
    android:textSize="30dp" />
```

```
<Button
    android:id="@+id/ViewAll"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_x="100dp"
    android:layout_y="500dp"
    android:text="View All"
    android:textSize="30dp" />
</AbsoluteLayout>
```

### **MainActivity.java**

```
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
public class MainActivity extends AppCompatActivity implements android.view.View.OnClickListener {
```

```

EditTextRollno,Name,Marks;
    Button Insert,Delete,Update,View,ViewAll;
    SQLiteDatabase db;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Rollno=(EditText)findViewById(R.id.Rollno);
        Name=(EditText)findViewById(R.id.Name);
        Marks=(EditText)findViewById(R.id.Marks);
        Insert=(Button)findViewById(R.id.Insert);
        Delete=(Button)findViewById(R.id.Delete);
        Update=(Button)findViewById(R.id.Update);
        View=(Button)findViewById(R.id.View);
        ViewAll=(Button)findViewById(R.id.ViewAll);

        Insert.setOnClickListener(this);
        Delete.setOnClickListener(this);
        Update.setOnClickListener(this);
        View.setOnClickListener(this);
        ViewAll.setOnClickListener(this);

        // Creating database and table
        db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS student(rollnoVARCHAR,nameVARCHAR,marks
        VARCHAR);");
    }
    public void onClick(View view)
    {
        // Inserting a record to the Student table
        if(view==Insert)
        {
            // Checking for empty fields
            if(Rollno.getText().toString().trim().length()==0 ||
            Name.getText().toString().trim().length()==0 ||
            Marks.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter all values");
                return;
            }
            db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+"'");

```

```

        ""+Marks.getText()+"");");
showMessage("Success", "Record added");
clearText();
    }
    // Deleting a record from the Student table
if(view==Delete)
    {
        // Checking for empty roll number
if(Rollno.getText().toString().trim().length()==0)
    {
        showMessage("Error", "Please enter Rollno");
return;
    }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'", null);
if(c.moveToFirst())
    {
        db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
        showMessage("Success", "Record Deleted");
    }
else
    {
        showMessage("Error", "Invalid Rollno");
    }
clearText();
    }
    // Updating a record in the Student table
if(view==Update)
    {
        // Checking for empty roll number
if(Rollno.getText().toString().trim().length()==0)
    {
        showMessage("Error", "Please enter Rollno");
return;
    }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'", null);
if(c.moveToFirst()) {
        db.execSQL("UPDATE student SET name='"+ Name.getText() + "',marks='"+ Marks.getText() +
            "" WHERE rollno='"+Rollno.getText()+"'");
        showMessage("Success", "Record Modified");
    }
else {
        showMessage("Error", "Invalid Rollno");
    }
clearText();

```

```

    }
    // Display a record from the Student table
    if(view==View)
    {
        // Checking for empty roll number
        if(Rollno.getText().toString().trim().length()==0)
        {
            showMessage("Error", "Please enter Rollno");
            return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'", null);
        if(c.moveToFirst())
        {
            Name.setText(c.getString(1));
            Marks.setText(c.getString(2));
        }
    }
    else
    {
        showMessage("Error", "Invalid Rollno");
        clearText();
    }
}
// Displaying all the records
if(view==ViewAll)
{
    Cursor c=db.rawQuery("SELECT * FROM student", null);
    if(c.getCount()==0)
    {
        showMessage("Error", "No records found");
        return;
    }
    StringBuffer buffer=new StringBuffer();
    while(c.moveToNext())
    {
        buffer.append("Rollno: "+c.getString(0)+"\n");
        buffer.append("Name: "+c.getString(1)+"\n");
        buffer.append("Marks: "+c.getString(2)+"\n\n");
    }
    showMessage("Student Details", buffer.toString());
}
}
public void showMessage(String title,String message)
{
    AlertDialog.Builder builder=new AlertDialog.Builder(this);

```

```

builder.setCancelable(true);
builder.setTitle(title);
builder.setMessage(message);
builder.show();
}
public void clearText()
{
Rollno.setText("");
Name.setText("");
Marks.setText("");
Rollno.requestFocus();
}
}

```

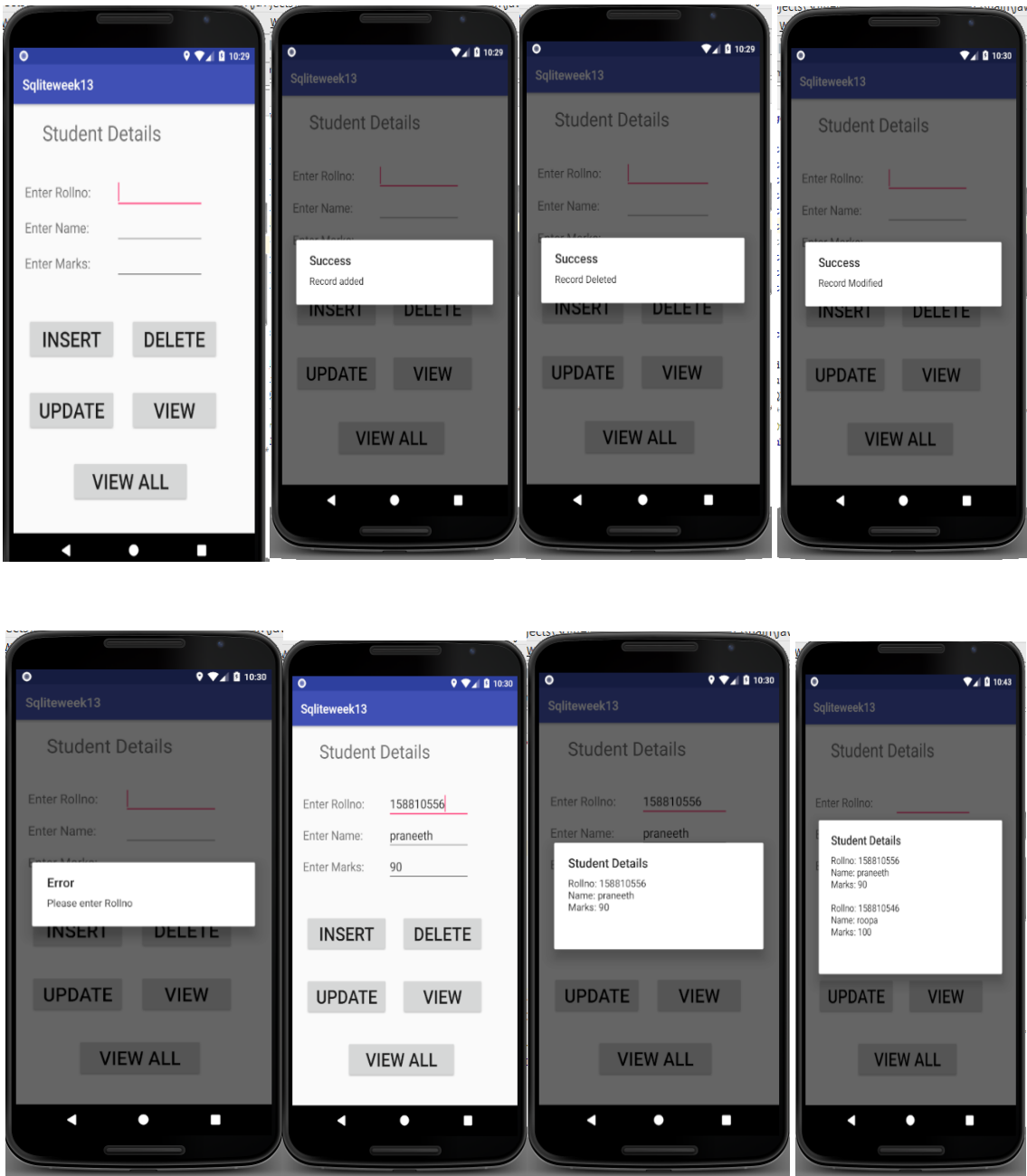
#### **Pre Lab VIVA Questions:**

- 1.What are different DataBase operations?
- 2.Write about SQLite DataBase.
3. Different methods used in SQLite DataBase?
- 4.Difference between execSQL() and.rawQuery().
5. What is the syntax of Button Component?
6. What is the use of cursor?
- 7.What does AndroidManifest.XML file contains?
- 8.What is the method used to set the text?

#### **Post Lab VIVA Questions :**

- 1.What is syntax used to create DataBase.
2. Write the code Snippet to retrieve the data from the DataBase.
- 3.What are the methods of Cursor class?
- 4.What is the use of StringBuffer class?
- 5.What is the usage of moveToFirst() in cursor?
- 6.What is the method used to fetch the data entered by user in EditText field?
- 7.Write the syntax for Select, Insert, Delete and Update operations?
- 8.Write the syntax to fetch the data from database?
- 9.Write the syntax to connect to a DataBase?

Output:





## VIVA QUESTIONS

### 1) What is Android?

**Answer:** Android is an open-source operating system and is mainly popular for Smartphones and Tablets.

This operating system is Linux Kernel-based. Using the Android operating system, the developer develops the functions or programs which can perform basic as well as the advanced type of operations on the Smartphone.

### 2) What is the Android SDK?

**Answer:** To develop a mobile application, Android developers require some tools and this requirement is satisfied by “Android SDK” which is a set of tools that are used for developing or writing apps.

It has a Graphical User Interface that emulates the Android environment. This emulator acts like an actual mobile device on which the developers write their code and then debug/test the same code to check if anything is wrong.

### 3) What are the different versions of Android OS that you remember?

**Answer:** Given below are the various versions of Android.

Versi on	Name
Android 8.0	Oreo
Android 7.0 – 7.1.2	Nougat
Android 6 – 6.0.1	Marshmallow
Android 5 – 5.1.1	Lollipop
Android 4.4 – 4.4.4	KitKat
Android 4.1 – 4.3	Jelly Bean
Android 4.0-4.0.4	Ice Cream Sandwich

#### 4) What is the difference between Mobile Application Testing and Mobile Testing?

**Answer:** Mobile app testing is the testing of applications on a device which mainly focuses on functions and features of the application.

And Mobile Testing is the testing of the actual mobile device and focuses on mobile features like Call, SMS, Contacts, Media Player, inbuilt browsers, etc.

#### 5) Name the languages supported for Android development.

**Answer:** Java is the widely used language for Android development.

It also supports C/C++ and when used with Android SDK, it improves the performance speed too.

#### 6) What are the advantages of the Android Operating System?

**Answer:** It is open-source and platform-independent. It supports various technologies like Bluetooth, Wi-Fi, etc

#### 7) Explain Android Architecture briefly.

**Answer:** Android architecture is in the form of software stack components.

**The below diagram describes the different layers in the Android architecture.**

- **Linux Kernel:** Linux Kernel is placed at the bottom of the software stack and is the foundation of the Android architecture. Using Linux kernel, Android provides a connection between the other layers of the software. It helps to develop drivers like the keypad, display, audio for device manufacture, etc.
- **Hardware Abstraction Layer (HAL):** HAL provides an interface between device drivers and API framework. It consists of library modules that are specific to the hardware component.
- **Android Runtime:** Linux kernel provides a multi-tasking execution environment so that multiple processes can execute each process runs on its own instance of Android Runtime (ART). Android has core runtime libraries like Dalvik VM specific libraries, Java Interoperability Libraries, Android Libraries, and C/C++ libraries.



### source

- **Application Framework (Java API Framework):** The entire android functionalities are available through the API. It consists of multiple services like Activity Manager, Resource Manager, Notification Manager, etc., which form the environment in which the android application runs.
- **Applications:** The Android application is a top layer and all types of in-built applications such as SMS, Browsers, Contact, etc are included in this top layer. It also includes third-party applications that are installed by the user such as Games, etc.

### **8) Define and explain the Android Framework.**

**Answer:** Android framework is a set of API's using which the Android developers write code for the mobile apps. It contains the methods and classes to write the programming code.

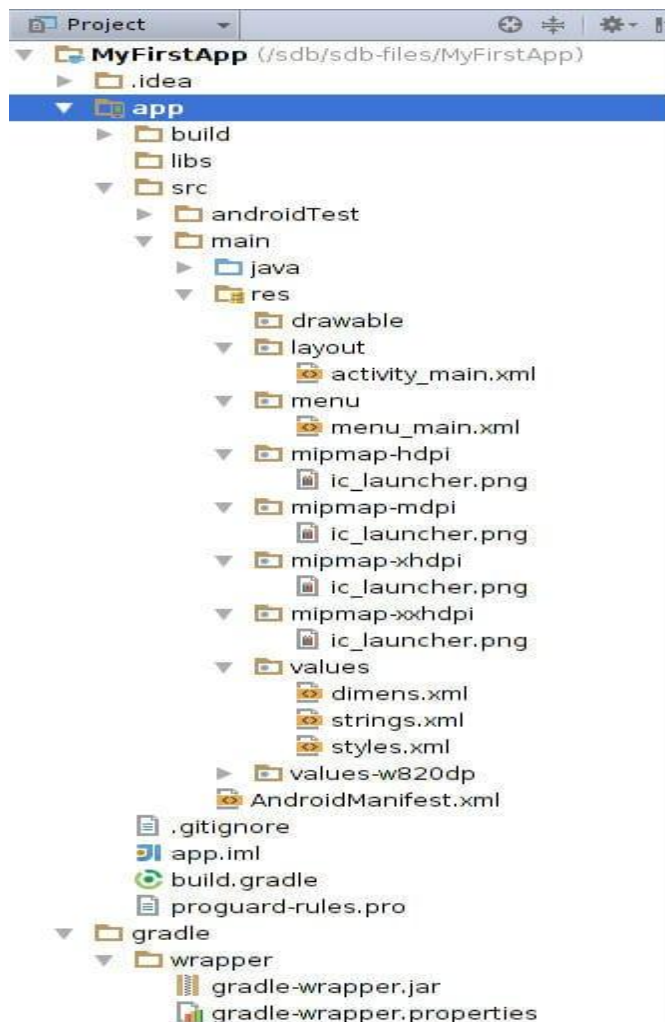
Android framework includes a different set of tools to create image pane, text field, buttons, etc. It also includes "Activities" with which the user interacts and "Services", which are the programs that run in the background. It is a package of different components like Intents, Broadcast Receivers, Content Providers, etc.

### 9) Which components are necessary for a New Android project?

**Answer:** Whenever a new Android project is created, the below components are required:

- **manifest:** It contains an **XML** file.
- **build/:** It contains build output.
- **src/:** It contains the code and resource files.
- **res/:** It contains bitmap images, UI Strings and XML Layout i.e. all non-code resources.
- **assets/:** It contains a file that should be compiled into a **.apk** file.

The below image shows the Project View once an Android project is created:



## 10) Provide the important core components of Android.

**Answer:** The core components of Android operating systems are:

- Activity
- Intents
- Services
- Content Provider
- Fragment

## 11) Explain briefly – what is meant by Activities?

**Answer:** Activities are the part of the mobile app which the user can see and interact with.

**For Example,** if you open an SMS app which has multiple activities like create new SMS, add a contact from the address book, write the content in the SMS body, send SMS to the selected contact, etc.

**Activity keeps a track of the following:**

- Keeps track of what a user is currently looking for in an app.
- Keeps a track of previously used processes, so that the user can switch between ongoing process and previous process.
- It helps to kill the processes so that the user can return to their previous state

**An activity is implemented as a subclass of Activity class as shown below:**

```
Public class MyActivity extends Activity
{
}
```

## 12) What is meant by Services?

**Answer:** Service is an Android component that runs in the background and acts independently. It does not provide any user interface.

Though the services are running behind the scene, a user can continue their work on different apps. Most of the time, the users are not aware of the services which are running in the background. These services allow the system to kill the process without interrupting the user's ongoing work.

**A service is implemented as a subclass of Service class:**

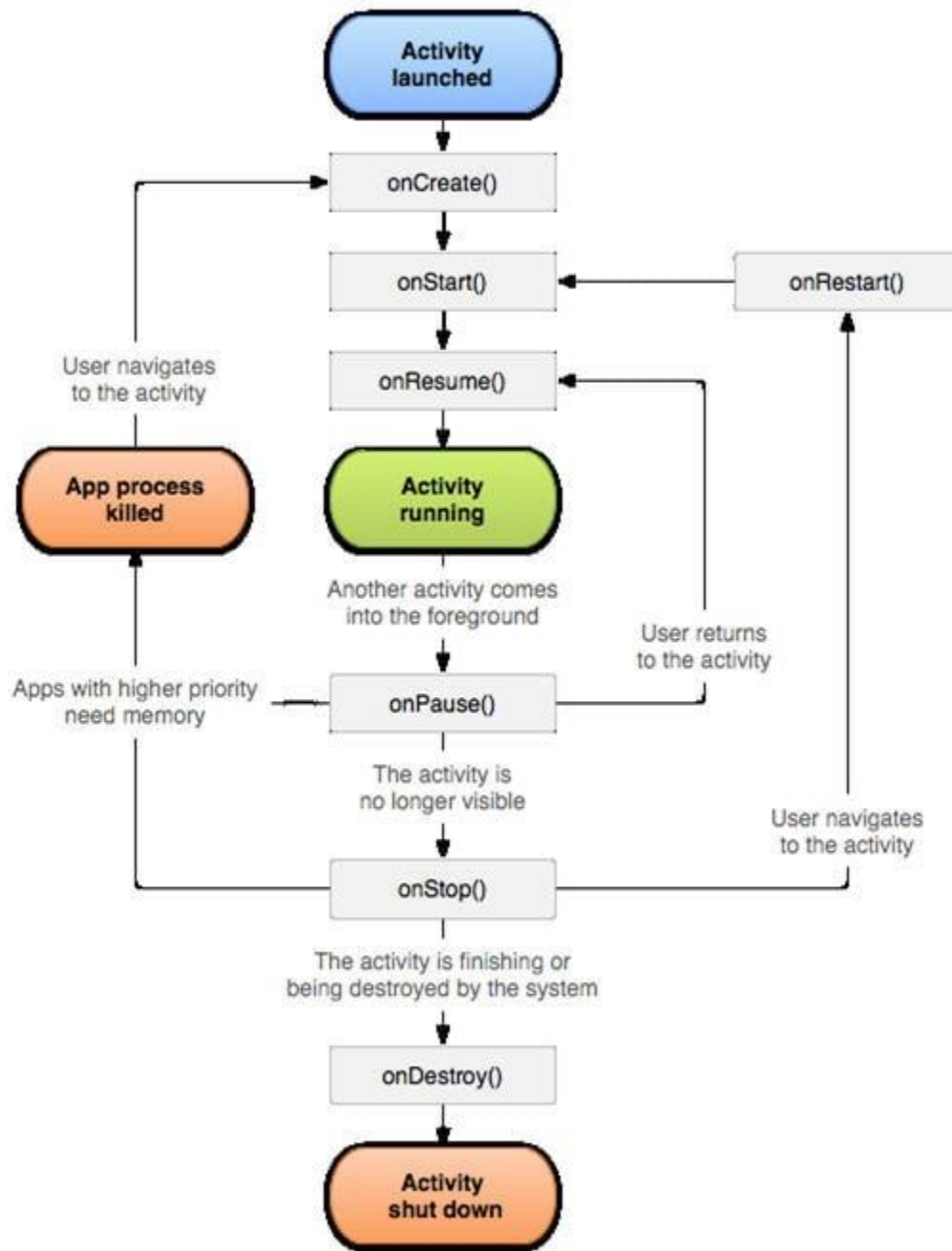
```
Public class MainService extends Service
{
}
```

**13) Explain Activity Lifecycle briefly.**

**Answer:** When a user interacts with the app and moves here and there, out of the app, returns to the app, etc. During all this process “Activity” instances also move in the different stages in their lifecycle.

There are seven different states like – onCreate(), onStart(), onRestart(), onResume(), onPause(), onStop(), and onDestroy(). These are termed as a ‘callback’. Android system invokes these callbacks to know that the state has been changed.

**The below-given diagram describes the Activity Lifecycle:**



When a user is working on an app, then there are many activities involved in it like Open, Close, Save, Delete, Send, etc.

Based on the user action these activities are partially disconnected from the UI but these activities always reside in the memory so that when the user calls back the same activity, the user will be in the same state where he has left off.

#### 14) What is an Intent?



**Answer:** Android has an Intent class when the user has to navigate from one activity to another. Intent displays notifications from the device to the user and then the user can respond to the notification if required.

**Given below are the two types:**

- Implicit Intents
- Explicit Intents

**15) Explain Implicit and Explicit Intents.**

**Answer:** Implicit Intent calls the system components while explicit Intents invoke the Activity class.

**16) What is the importance of setting up permission in app development?**

**Answer:** Once the permissions are set for the app development, then the data and code are restricted to the authorized users only.

If the code is kept without any restriction or if it is accessible to anyone then there are chances of compromise of code which results in defect leakage.

**17) What is .apk extension in Android?**

**Answer:** It is a default file format that is used by the Android Operating System. Application Package Kit (APK) is used for the installation of mobile apps. The .apk contains resource file, certificate, manifest file, and other code.

APK files are archive files in the zip format with .apk extension.

**18) What is the database used for the Android platform?**

**Answer:** SQLite is the database that is used for the Android platform. It is an open-source, serverless database.

### 19) What is ANR in Android?

**Answer:** ANR stands for Application Not Responding. It is a notification or pop-up displayed by the Android platform whenever the application is performing too many functions at a time and if it is suddenly not responding for a long time to the user action.

### 20) Which are the dialog boxes supported by the Android platform?

**Answer:** Android supports four types of dialog boxes:

- **AlertDialog:** It has a maximum of 3 buttons and sometimes AlertDialog includes checkboxes and Radio buttons to select the element.
- **ProgressDialog:** It displays the progress bar or wheels.
- **TimePickerDialog:** Using this dialog box, a user selects the Time.
- **DatePickerDialog:** Using this dialog box, a user selects the Date

### 21) What is ADB?

**Answer:** Android Debug Bridge (ADB) is a command-line tool that performs shell commands.

ADB is used for direct communication between the emulator ports. It gives direct control of the communication between the emulator instances to the developer.

### 22) What is ActivityCreator?

**Answer:** ActivityCreator is a batch file and shell script which was used to create a new Android project. It is now replaced by the "Create New Project" in Android SDK.

### 23) What is Orientation?

**Answer:** Orientation is the key feature in Smartphones nowadays. It has the ability to rotate the screen between Horizontal or Vertical mode.

**Android supports two types of screen Orientations as mentioned below:**

- **Portrait:** When your device is vertically aligned.
- **Landscape:** When your device is horizontally aligned.

setOrientation() is a method using which you can set a screen alignments. HORIZONTAL and VERTICAL are two values that can be set in the setOrientation() method. Whenever there is a change in the display orientation i.e. from Horizontal to Vertical or vice versa then onCreate() method of the Activity gets fired.

Basically, when the orientation of the Android mobile device gets changed then the current activity gets destroyed and then the same activity is recreated in the new display orientation. Android developers define the orientation in the AndroidManifest.xml file.

#### **24) What is AIDL?**

**Answer:** In the Android platform, there are remote methods that facilitate the use of methods from one program to another. To create and implement the remote methods the first step is to define the communication interface in AIDL.

AIDL stands for Android Interface Definition Language. It facilitates communication between the client and the service. It also communicates the information through inter-process communication.

For communication between processes, the data is broken down into chunks which are easily understandable by the Android platform.

#### **25) What are the data types supported by AIDL?**

**Answer: Data Types supported by AIDL are as follows:**

- String
- List
- Map
- CharSequence
- Java data types such as INT, Long, Char, Boolean, etc

#### **26) Explain the AndroidManifest.xml file and why do you need this?**

**Answer:** Every application must have an AndroidManifest.xml file in the root directory. It contains information about your app and provides the same to the Android system.

The information includes the package name, Android components such as Activity, Services, Broadcast Receivers, Content Providers, etc. Every Android system must have this information before running any app code.

**AndroidManifest.xml file performs the following tasks:**

- It provides a name to the Java package and this name is a unique identifier for the application.
- It describes the various components of the application which include Activity, Services, Content Providers, etc. Also, it defines the classes which implement these components.
- It is responsible to protect the application and it declares the permission for accessing the protected part of the app.
- It also declares the Android API which is going to be used by the application.

- It contains the library file details which are used and linked to the application.

**27) What all devices have you worked on?**

**Answer:** There are many mobile devices available in the market with different operating systems.

Specifically, I have worked on Android, Windows, Symbian, iPhone, etc

**28) Which tools are used for debugging on the Android platform?**

**Answer:** To understand the cause of the failure or cause of any issue, debugging is important. On the Android platform **Android Monitor.bat** utility is used while on the iOS platform, iPhone Configuration utility is used for debugging purposes.

There are different tools for debugging which include – Android DDMS, Android Debug Bridge, iOS simulator, Debugging from Eclipse with ADT, Remote debugging on Android with Chrome, etc.

**29) Which scenario can test only on real devices but not on an emulator?**

**Answer:** Emulators are used for performing similar kinds of testing which is performed on the real devices. Basically, emulators are used as a replacement for real devices as sometimes real devices are not available for testing, the use of real mobile devices for testing purposes is costlier at times.

But there are few scenarios that cannot be tested using emulator, these can be tested only using real devices. These scenarios are interrupted scenarios i.e. message, phone call interruption while using the app, low battery, Bluetooth, memory card mount and unmount, etc.

**30) Name the mobile automation tools that are available in the market.**

**Answer:** There are quite a few mobile automation testing tools that are available in the market but these are used only if the project requires it and if the application supports the automation.

These tools are paid as well as free tools, hence analysis needs to be done within the project team and then the appropriate mobile automation tool needs to be selected. Silk Mobile, SeeTest, Ranorex are the paid mobile automation tool while Appium, KIF, Robotium, Calabash are few free tools.

### 31) How do you troubleshoot the android application which is crashing frequently?

**Answer:** Given below are the few steps that we need to follow while troubleshooting the crashing issue:

- **Free up memory space:** There is only limited space available on mobile devices for mobile apps. To avoid crashing issues or memory-related issues, you need to first check the memory space.
- **Clear app data usage:** You can clear the app data using the Application Manager under "Settings". This will clear the cache memory and allow some free space to install another app or it will boost up your current app.
- **Memory Management:** Some apps run perfectly on one type of mobile device but the same app may not work on another type of device as for such devices the processing power, memory management, and CPU speed is different. For any app to run properly on any type of mobile device, you should manage the memory on the device.
- **Compatibility issue:** It is always not possible to test mobile apps on all mobile devices, browsers, operating systems, etc. So you need to test your mobile app on as many mobile devices as you can in order to avoid any compatibility issue.

### 32) How do you find memory leaks in the mobile app on the Android platform?

**Answer:** Android Studio is using Android Device Manager (ADM), this ADM is used to detect the memory leaks in the Android platform.

When you open ADM in the Android Studio then on the left-hand side of the ADM, you will find your device or emulator in which a heap sign will be displayed. When you are running any mobile app then you will see the heap size, memory analysis and other statistics displayed on it.

### 33) What is DDMS?

**Answer:** Android Studio has debugging tools known as DDMS i.e. Dalvik Debug Monitor Server.

**It has wide debugging features which include:**

- Port forwarding services.
- Screen capture on the device.
- Thread and Heap information.
- Incoming call and SMS spoofing.
- Logcat
- Radio state information.
- Location data spoofing.

DDMS is integrated with the Android studio. To launch the DDMS, you need to open the Android Device Monitor (ADM) first and then click on the DDMS menu button. Once DDMS is launched, then on the left-hand side the list of connected devices is displayed along with the processes which are running on each device.

With the help of DDMS, you can debug both on real devices and emulators.

### **34) What are the different data storage options available on the Android platform?**

**Answer:** Android platform provides a wide range of data storage options. These options must be used based on the need such as data is secure and used with permission only or can be accessed publicly.

**Below is the list of data storage options on the Android platform:**

- **SharedPreferences:** It stores data in XML files. It is the simplest way to store private data in the key-value pair.
- **SQLite:** It stores structured data in the private database.
- **Internal Storage:** It stores data in the device file system and any other app cannot read this data.
- **External Storage:** Data is stored in the file system but it is accessible to all apps in the device

## WORKSHEET



## WORKSHEET

## WORKSHEET

## WORKSHEET

## WORKSHEET

## WORKSHEET