



KARPAGAM ACADEMY OF HIGHER EDUCATION

Karpagam University

(Deemed University Established Under Section 3 of UGC Act 1956)

COIMBATORE – 641 021

DEPARTMENT OF COMPUTER SCIENCE, CA & IT

SUBJECT: Android Programming - Practical

SEMESTER: III

L T P

SUBJECT CODE: 16CSU314A

CLASS: II B.Sc.CS

0 0 3

LIST OF PRACTICALS

1. Create —Hello World application that will display —Hello World in the middle of the screen in the emulator. Also display —Hello World in the middle of the screen in the Android Phone.
2. Create an application with login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and selected option should appear in text box.
5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.
6. Create an application with three option buttons, on selecting a button colour of the screen will change.
7. Create and Login application as above. On successful login, pop up the message.
8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.

ESE MARKS ALLOCATION

1	Program I	20
2	Program II	20
3	Viva-voce	10
4	Record	10
Total		60

CONTENTS

Ex. No	NAME OF THE EXPERIMENT
1.	Creation of Hello World Application
2.	Application With Login Module
3.	Creating a Spinner With Strings
4.	Creating a Menu Application
5.	Creating an Application to List Courses of the College
6.	Creating an Option button to Change the Color of the Screen
7.	Creating a Login Page and displaying the Pop up Message
8.	Creating an Application to Create, Insert, Update, Delete and Retrieve Operation on the Database

Ex.No :1

Creation of Hello World Application

Aim: To develop an Android Application “Hello World” with common activity.

Algorithm:

Step 1: Open Eclipse IDE

Step2: Select New Android Application Project

Step3: Create a Main Activity Java Program

Step4: Create a Main Activity XML Layout for Android Interface

Step5: Open Android Virtual Device Manager

Step6: Run the project as Android Application project

Step7: Hello World App is displayed on a Android App Emulator

MainActivity.java Program

```
package com.example.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends Activity {

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

```

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

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

Activity_main.xml file

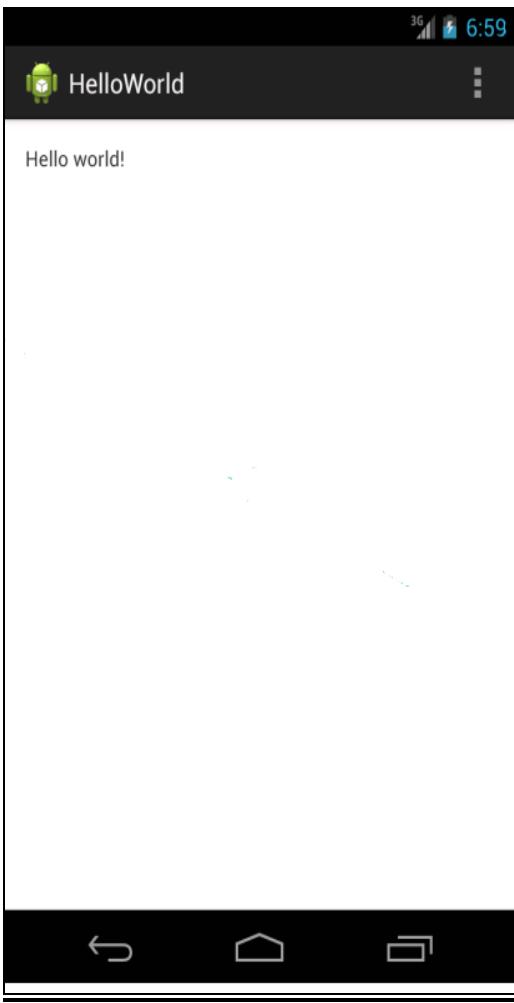
```

<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="com.example.helloworld.MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
</RelativeLayout>

```

OUTPUT



Result:

HelloWorld Android App is executed successfully.

Ex No: 2.

Creating an Application with Login Module

Aim: To develop a login screen in Android Application

Algorithm:

Step 1: Open Eclipse IDE

Step 2: Select New Android Application Project

Step 3: Create a Main Activity XML Layout for Android Interface

Step 4: Drag and drop plain text field for entering loginid

Step 5: Drag and drop password text field for entering password

Step 6: Create one button login

Step 7: Write a Java coding for matching loginid and password and display validation message

Step 8: Open Android Virtual Device Manager

Step 9: Run the project as Android Application project

Step 10: Login App is executed on a Android App Emulator

Main Activity.java program

```
package com.example.exp23;
import android.app.Activity;
import android.os.Bundle;
import android.view.*;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.*;
```

```

import android.view.View.OnClickListener;

public class MainActivity extends Activity implements OnClickListener{

    Button b = null;
    EditText et1 = null;
    EditText et2 = null;
    TextView tv = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b=(Button)findViewById(R.id.button1);
        et1=(EditText)findViewById(R.id.username);
        et2=(EditText)findViewById(R.id.password);
        tv=(TextView)findViewById(R.id.my_TextView);
        b.setOnClickListener(this);  }

    public void onClick(View v){
        String temp1 =et1.getText().toString();
        String temp2 =et2.getText().toString();
        if(temp1.equalsIgnoreCase(temp2))
            tv.setText("VALID");
        else
            tv.setText("INVALID");  }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
```

```
        return true;
    }
    return super.onOptionsItemSelected(item);  } }
```

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="com.example.exp23.MainActivity" >

    <TextView
        android:id="@+id/my_TextView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="My Login Form application" />

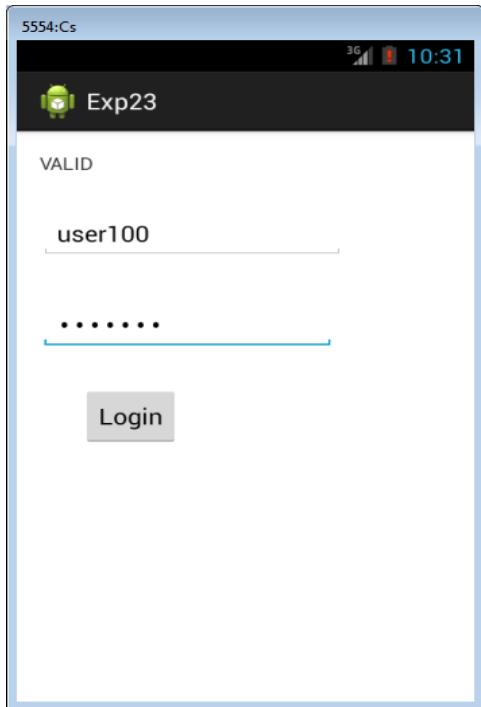
    <EditText
        android:id="@+id/username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/my_TextView"
        android:layout_below="@+id/my_TextView"
        android:layout_marginTop="28dp"
        android:ems="10" >

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/username"
        android:layout_centerVertical="true"
        android:layout_marginLeft="29dp"
        android:text="Login" />
```

```
<EditText  
    android:id="@+id/password"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignLeft="@+id/username"  
    android:layout_below="@+id/username"  
    android:layout_marginTop="34dp"  
    android:ems="10"  
    android:inputType="textPassword" />  
</RelativeLayout>
```

OUTPUT



Result:

Login Android App for displaying validation message is executed successfully.

Ex No: 3

Creating a Spinner with Strings

Aim: To implement a spinner control in Android Application

Algorithm:

Step 1: Open Eclipse IDE

Step 2: Select New Android Application Project

Step 3: Create a Main Activity XML Layout for Android Interface

Step 4: Drag and drop spinner control

Step 5: Drag and drop plain text field for displaying output.

Step 6: Write a Java coding for selecting options from spinner control and displaying the selected option in the text view control

Step 7: Open Android Virtual Device Manager

Step 8: Run the project as Android Application project

Step 9: Spinner App is executed on a Android App Emulator

MainActivity.java Program

```
package com.example.spinw;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
```

```

import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity implements
AdapterView.OnItemSelectedListener {
    TextView tv = null;

    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.spinner1);
        spin.setOnItemSelectedListener(this);
        tv=(TextView)findViewById(R.id.textView1);
        //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);
        tv.setText("");
    }

    //Performing action onItemSelected and onNothing selected
    @Override
    public void onItemSelected(AdapterView<?> arg0, View arg1, int position,long id) {
        tv.setText("");
        Toast.makeText(getApplicationContext(),country[position],
            Toast.LENGTH_LONG).show();
        tv.setText(country[position]);
    }

    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        // TODO Auto-generated method stub
    }
}

```

Activity_main.xml Program

```
<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="com.example.spinw.MainActivity" >

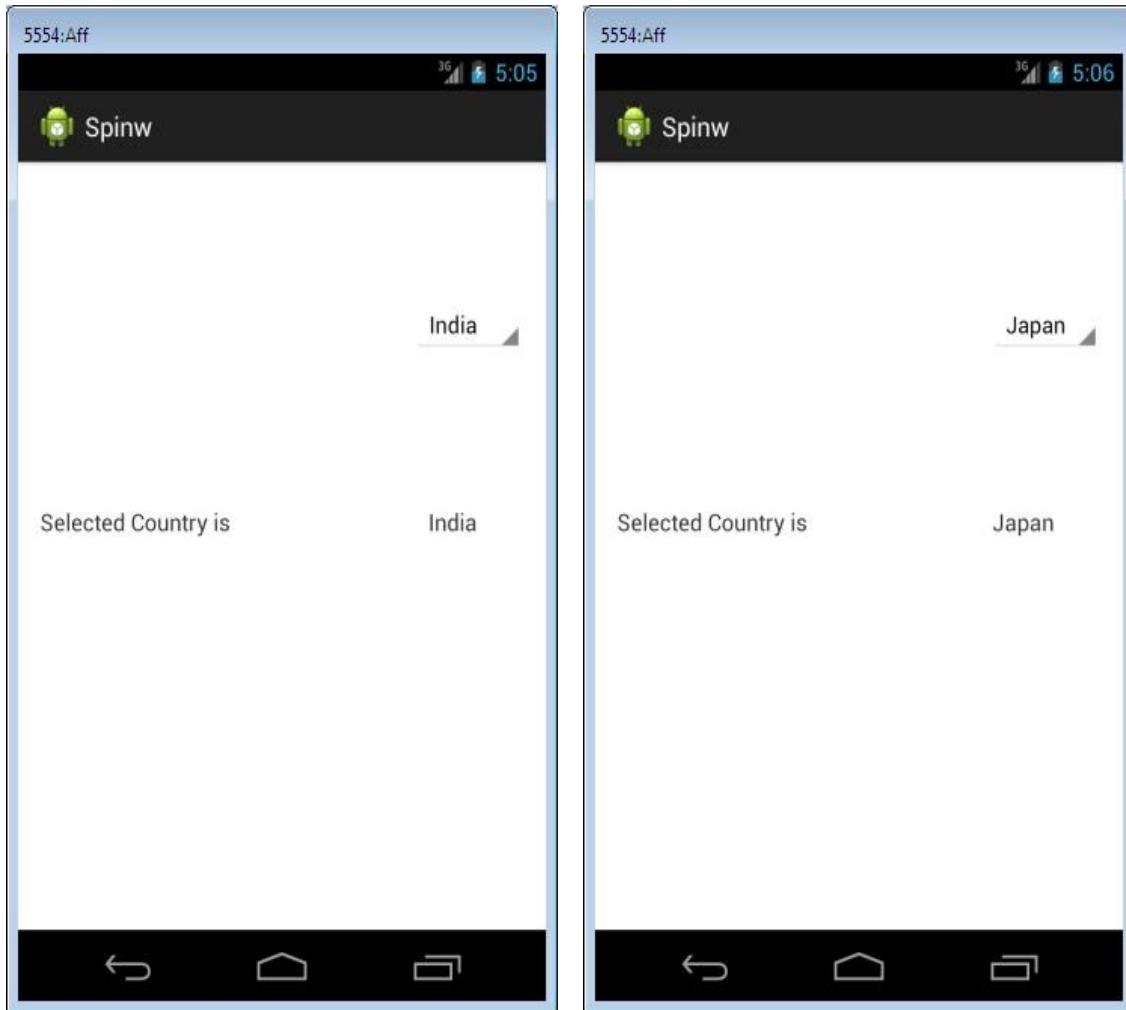
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textView1"
        android:layout_alignBottom="@+id/textView1"
        android:layout_alignParentLeft="true"
        android:text="Selected Country is"
        android:textSize="16dp" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/spinner1"
        android:layout_below="@+id/spinner1"
        android:layout_marginRight="34dp"
        android:layout_marginTop="104dp"
        android:text="@string/hello_world"
        android:textSize="16dp" />

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="76dp" />
```

</RelativeLayout>

OUTPUT:



Result:

Spinner control Android App is executed successfully.

Ex No: 4

Creating a Menu Application

Aim: To develop an Android Application for implementing Sub Menu.

Algorithm:

Step 1: Open Eclipse IDE

Step 2: Select New Android Application Project

Step 4: Create a Main Activity XML Layout for Android Interface

Step 5: Create Sub Menu using Main.xml file

Step 6: Create a Main Activity Java Program

Step 7: Write a Java coding for Menu selecting operations

Step 5: Open Android Virtual Device Manager

Step 6: Run the project as Android Application project

Step 7: Sub Menu App is executed on a Android App Emulator

Main Activity.java program

```
package com.example.sumenu1;

import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;

public class MainActivity extends Activity {

    @Override
```

```

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

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

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    TextView text = (TextView) findViewById(R.id.Hello);
    if (id == R.id.red) {
        text.setTextColor(Color.RED);
    }
    if (id == R.id.blue) {
        text.setTextColor(Color.BLUE);
    }
    if (id == R.id.yellow) {
        text.setTextColor(Color.YELLOW);
    }
    return super.onOptionsItemSelected(item);
}
}

```

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="com.example.kopw.MenuActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
</RelativeLayout>

```

Main.xml for creating Menus

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.example.sumenu1.MainActivity" >

    <item
        android:id="@+id/red"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="@string/red"/>
    <item
        android:id="@+id/blue"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="@string/blue"/>
    <item
        android:id="@+id/yellow"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="@string/yellow"/>

</menu>

```

String.xml for setting values to the Menu

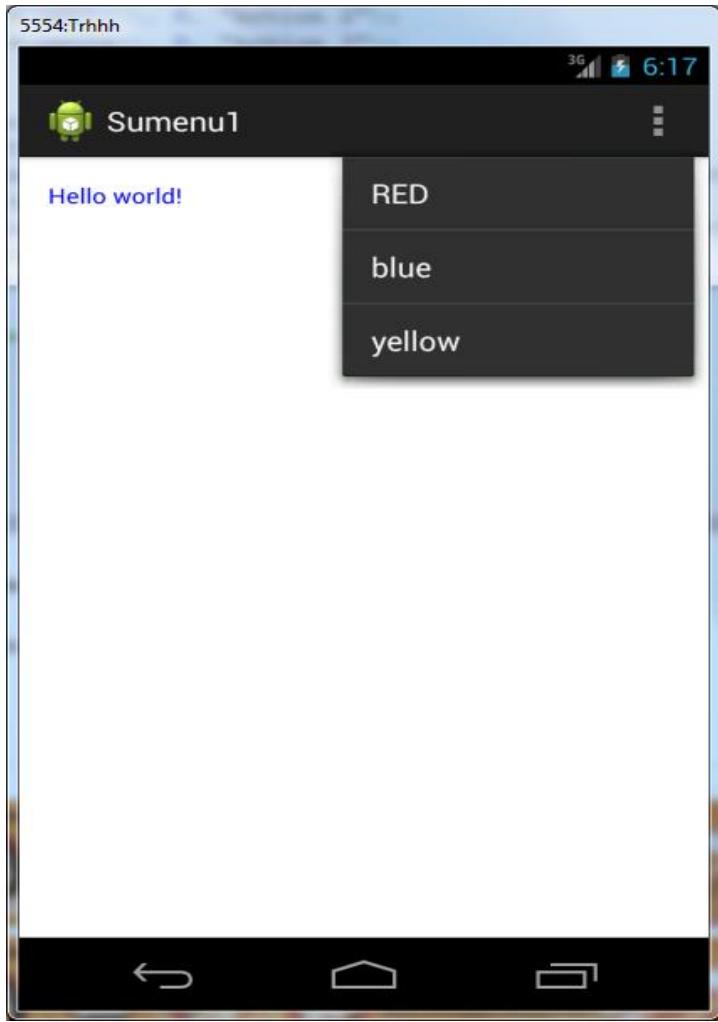
```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Sumenu1</string>
    <string name="hello_world">Hello world!</string>
    <string name="red">RED</string>
    <string name="blue">blue</string>

```

```
<string name="yellow">yellow</string></resources>
```

OUTPUT



Result :

Sub Menu Android App is executed successfully.

Ex No: 5

Creating an Application to List Courses of the College

Aim: To implement a program with list of courses in your college and display the in-charge of the course at the bottom of the screen

Algorithm:

Step 1: Open Eclipse IDE

Step 2: Select New Android Application Project

Step 3: Create a Main Activity XML Layout for Android Interface

Step 4: Drag and drop list control

Step 5: Add the list of courses available in the Karpagam University in the list control

Step 6: Write a Java coding for selecting options from list control and displaying the respective HOD for the course in the text control

Step 7: Open Android Virtual Device Manager

Step 8: Run the project as Android Application project

Step 9: List Control App is executed on a Android App Emulator

MainActivity.java Program

```
package com.example.list;
import android.app.Activity;
import android.os.Bundle;
import android.widget.*;
import android.view.View;
import android.widget.ListView;
import android.widget.AdapterView.OnItemClickListener;
```

```
public class MainActivity extends Activity {
    ListView listView ;
```

```

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

    // Get ListView object from xml
    listView = (ListView) findViewById(R.id.list);

    // Defined Array values to show in ListView
    String[] values = new String[] { "BCA Course ",
        "B.Sc CS Course ",
        "B.Sc IT Course "

    };

    // Define a new Adapter
    // First parameter - Context
    // Second parameter - Layout for the row
    // Third parameter - ID of the TextView to which the data is written
    // Forth - the Array of data

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, android.R.id.text1, values);

    // Assign adapter to ListView
    listView.setAdapter(adapter);

    // ListView Item Click Listener
    listView.setOnItemClickListener(new OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> parent, View view,
            int position, long id) {

            // ListView Clicked item index
            int itemPosition = position;

            // ListView Clicked item value
            String itemValue = (String) listView.getItemAtPosition(position);
            if (itemPosition==0)
            {

```

```

        // Show Alert
        Toast.makeText(getApplicationContext(),
            "Position :" +itemPosition+" ListItem : " +itemValue + "HOD is Dr.Sheeja
Madam", Toast.LENGTH_LONG)
            .show();
        }
        if (itemPosition==1)
        {
        // Show Alert
        Toast.makeText(getApplicationContext(),
            "Position :" +itemPosition+" ListItem : " +itemValue + "HOD is Dr.Veni
Madam", Toast.LENGTH_LONG)
            .show();
        }
        if (itemPosition==2)
        {
        // Show Alert
        Toast.makeText(getApplicationContext(),
            "Position :" +itemPosition+" ListItem : " +itemValue + "HOD is Dr.Guna
Madam", Toast.LENGTH_LONG)
            .show();
        }
    }

});
```

}

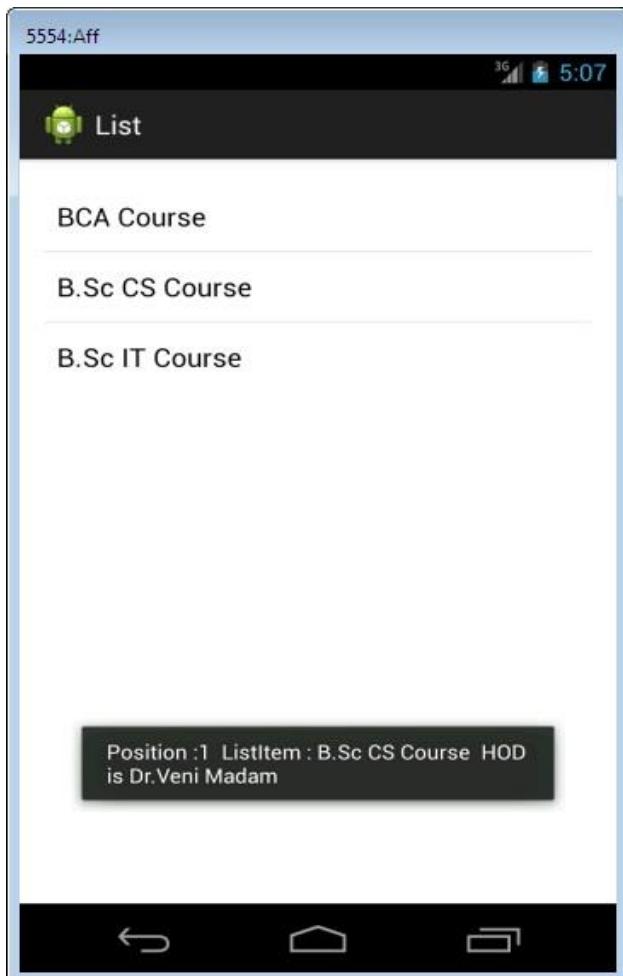
Activity_main.xml Program

```

<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="com.example.list.MainActivity" >
    <ListView
```

```
    android:id="@+id/list"
    android:layout_height="wrap_content"
    android:layout_width="match_parent">
</ListView></RelativeLayout>
```

OUTPUT:



Result:

List control Android App for displaying courses in University and displaying respective HOD of the course is executed successfully.

Ex No: 6

Creating an Option button to Change the Color of the Screen

Aim: To implement Radio button controls in Android Application

Algorithm:

Step 1: Open Eclipse IDE

Step 2: Select New Android Application Project

Step 3: Create a Main Activity XML Layout for Android Interface

Step 4: Drag and drop three radio button controls and put text in the three radio buttons as red, yellow and blue

Step 5: Write a Java coding for changing background color of the screen, when the respective radio button is clicked

Step 6: Open Android Virtual Device Manager

Step 7: Run the project as Android Application project

Step 8: Radio Button App is executed on a Android App Emulator

MainActivity.java Program

```
package com.example.rad1;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.graphics.Color;
import android.view.View;
import android.view.View.OnClickListener;
```

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

```

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final LinearLayout ll=(LinearLayout) findViewById(R.id.LinearLayout);

    final RadioButton radio_red = (RadioButton) findViewById(R.id.radio_red);
    final RadioButton radio_yellow = (RadioButton) findViewById(R.id.radio_yellow);
    final RadioButton radio_blue = (RadioButton) findViewById(R.id.radio_blue);
    radio_red.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {
            ll.setBackgroundColor(Color.RED);
        }
    });

    radio_yellow.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {
            ll.setBackgroundColor(Color.YELLOW);
        }
    });
    radio_blue.setOnClickListener(new OnClickListener() {

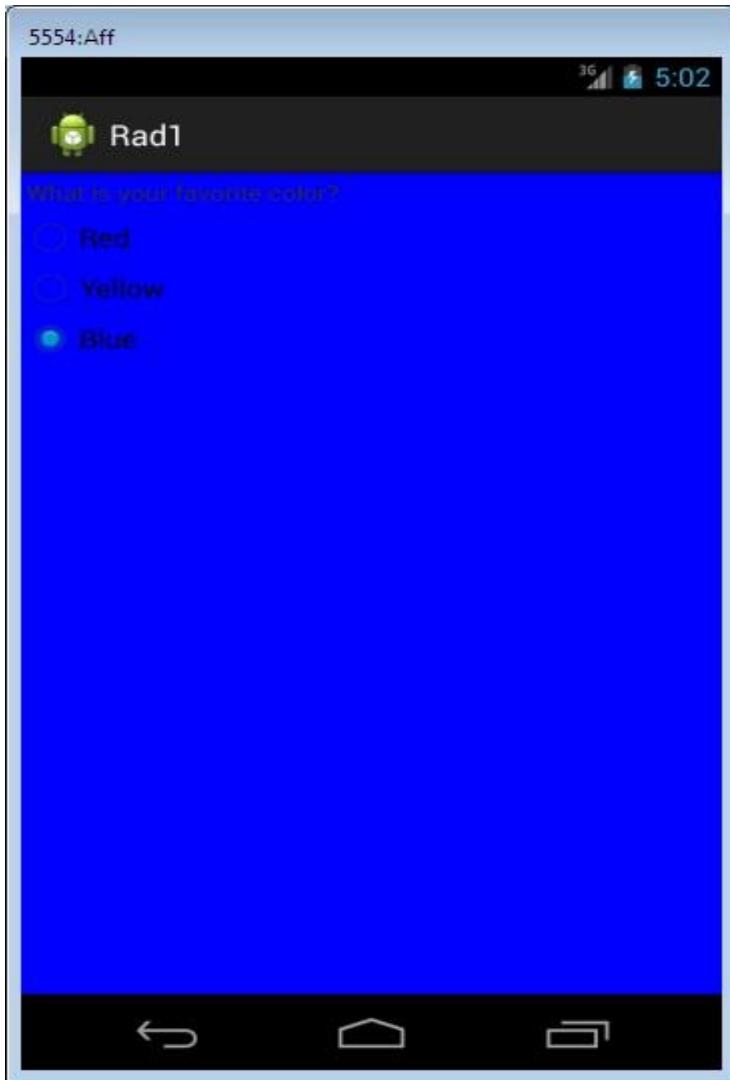
        public void onClick(View v) {
            ll.setBackgroundColor(Color.BLUE);
        }
    });
}

```

Activity_main.xml Program

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:layout_width="fill_parent"
    android:orientation="vertical"
    android:id="@+id/LinearLayout"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="fill_parent">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="What is your favorite color?"
    android:padding="3dip"/>
<RadioGroup android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/Group1"
    android:orientation="vertical">
<RadioButton android:id="@+id/radio_red"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Red"
    />
<RadioButton android:id="@+id/radio_yellow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Yellow" />
<RadioButton android:id="@+id/radio_blue"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Blue" />
</RadioGroup></LinearLayout>
```

OUTPUT:



Result:

Radio Button Android App for changing background color is executed successfully.

Ex No: 7

Creating a Login Page and displaying the Pop up Message

Aim: To implement a login module to display popup message

Algorithm:

Step 1: Open Eclipse IDE

Step 2: Select New Android Application Project

Step 3: Create a Main Activity XML Layout for Android Interface

Step 4: Drag and drop plain text field for entering loginid

Step 5: Drag and drop password text field for entering password

Step 6: Create one button login

Step 7: Write a Java coding for matching loginid and password and display validation message in the alert dialog box

Step 8: Open Android Virtual Device Manager

Step 9: Run the project as Android Application project

Step 10: Login App with alert dialogbox is executed on a Android App Emulator

MainActivity.java Program

```
package com.example.pop1;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.*;
```

```

import android.view.View.OnClickListener;

import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.content.DialogInterface;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity implements OnClickListener{

    final Context context = this;

    Button b = null;
    EditText et1 = null;
    EditText et2 = null;
    TextView tv = null;

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

        b=(Button)findViewById(R.id.button1);
        et1=(EditText)findViewById(R.id.username);
        et2=(EditText)findViewById(R.id.password);
        tv=(TextView)findViewById(R.id.my_TextView);
        b.setOnClickListener(this);
    }

    public void onClick(View v){
        String temp1 =et1.getText().toString();
        String temp2 =et2.getText().toString();
        if(temp1.equalsIgnoreCase(temp2))
        {
            AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(
                context);

            // set title
            alertDialogBuilder.setTitle("LOGIN is Valid!");
            AlertDialog alertDialog = alertDialogBuilder.create();
        }
    }
}

```

```

        // show it
        alertDialog.show();      }

else
{
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(
        context);
    // set title
    alertDialogBuilder.setTitle("LOGIN is INValid!");
    AlertDialog alertDialog = alertDialogBuilder.create();

    // show it
    alertDialog.show();

}
}

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

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

Activity_main.xml Program

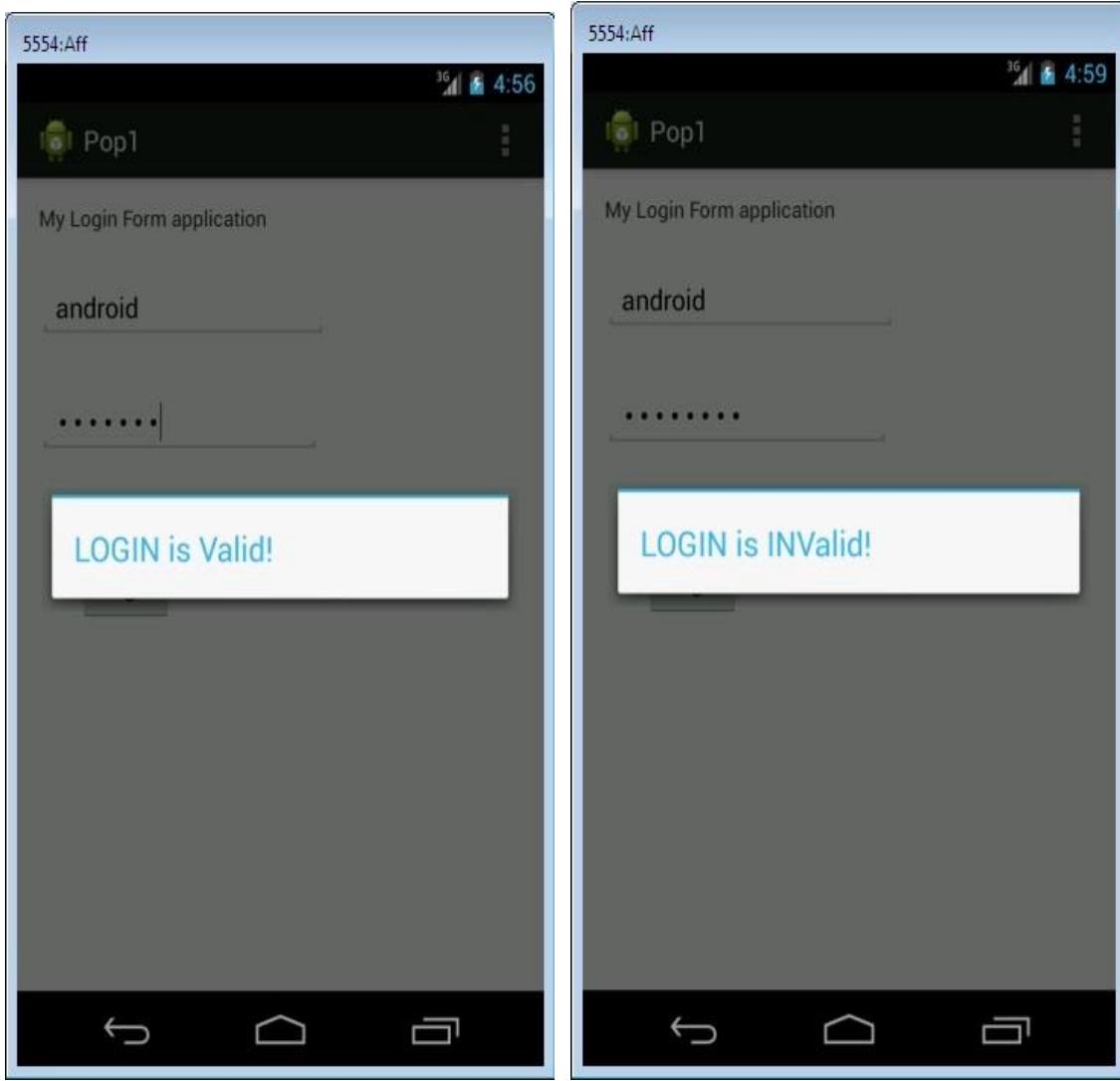
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.exp23.MainActivity" >
    <TextView
        android:id="@+id/my_TextView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="My Login Form application" />
    <EditText
        android:id="@+id/username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/my_TextView"
        android:layout_below="@+id/my_TextView"
        android:layout_marginTop="28dp"
        android:ems="10" >
        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/username"
        android:layout_centerVertical="true"
        android:layout_marginLeft="29dp"
        android:text="Login" />

    <EditText
        android:id="@+id/password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/username"
        android:layout_below="@+id/username"
        android:layout_marginTop="34dp" />
```

```
    android:ems="10"
    android:inputType="textPassword" />
</RelativeLayout>
```

OUTPUT:



Result:

Login Android App for displaying validation pop up message is executed successfully.

Ex No: 8

Creating an Application to Create, Insert, Update, Delete and Retrieve Operation on the Database

Aim: To implement a login module to display popup message

Algorithm:

Step 1: Open Eclipse IDE

Step2: Select New Android Application Project

Step 3: Create a Main Activity XML Layout for Android Interface

Step 4: Drag and drop plain text field for entering name

Step 5: Drag and drop text field for entering address

Step 6: Create one button to add the details in the database

Step 7: Write a Java coding for matching name and address and display message in the alert dialog box

Step 8: Open Android Virtual Device Manager

Step 9: Run the project as Android Application project

Step 10: Database App with alert dialogbox is executed on a Android App Emulator

MainActivity.java Program

```
package com.example.sqllite1;

import android.content.Context;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity implements View.OnClickListener {

    private EditText editTextName;
    private EditText editTextAdd;
    private Button btnAdd;
    private Button btnView;

    private SQLiteDatabase db;

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

        createDatabase();

        editTextName = (EditText) findViewById(R.id.editTextName);
        editTextAdd = (EditText) findViewById(R.id.editTextAddress);

        btnAdd = (Button) findViewById(R.id.btnAdd);
        btnAdd.setOnClickListener(this);

    }

    protected void createDatabase(){

    }
}
```

```
    db=openOrCreateDatabase("PersonDB", Context.MODE_PRIVATE, null);
    db.execSQL("CREATE TABLE IF NOT EXISTS persons(id INTEGER PRIMARY
KEY AUTOINCREMENT NOT NULL, name VARCHAR,address VARCHAR);");
}
```

```
protected void insertIntoDB(){
    String name = editTextName.getText().toString().trim();
    String add = editTextAdd.getText().toString().trim();
    if(name.equals("") || add.equals("")){
        Toast.makeText(getApplicationContext(), "Please fill all fields",
Toast.LENGTH_LONG).show();
        return;
    }

    String query = "INSERT INTO persons (name,address) VALUES('" +name+ "' ,
'" +add+"');";
    db.execSQL(query);
    Toast.makeText(getApplicationContext(), "Saved 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.main, menu);
    return true;
}
```

```
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
```

```
    int id = item.getItemId();
```

```
    //noinspection SimplifiableIfStatement
```

```
    if (id == R.id.action_settings) {
        return true;
    }
```

```
    return super.onOptionsItemSelected(item);
```

```

    }

@Override
public void onClick(View v) {
    if(v == btnAdd){
        insertIntoDB();
    }
}

```

Activity_main.xml Program

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:orientation="vertical"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name"
        android:id="@+id/textViewName" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editTextName" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Address"
        android:id="@+id/textViewAddress" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```
        android:id="@+id/editTextAddress" />

<Button
    style="?android:attr/buttonStyleSmall"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Add"
    android:id="@+id	btnAdd" />

</LinearLayout>
```

OUTPUT:



Result:

Database Android App for storing name and address is executed successfully