Exercise 1:

Aim:

To develop an Android application to display the given text. (for example. Hello World)

Procedure:

1. Open adk Bundle software.
2. Open Eclipse Folder and open eclipse .exe file
3. From File Menu, Choose New -> Android Application Project
4. Give hello as the project name
5. Click Next and Choose the Blank Activity and Click Finish
6. In the explorer window choose the res folder and choose layout
7. Double click Layout and open the activity_main.xml file by double clicking the file
8. From the bottom Tab, Choose Graphical Layout.
9. From the pallete in the left side of the graphical window drag a Large TextView control and place it in the activity window.
10. Similarly, drag a button control from the pallete and place it in the activity window.
11. Right Click on the button and choose Edit Text and click New String
12. Give 'Display' as String name and display as R.New String name
13. Right Click on the TextView control and give R.New String name as hello.
14. Right Click on the src folder in the explorer window and choose the MainActivity.java file
15. Double click the MainActivity.java file and write the following in the file in the public class MainActivity area.

```
public class MainActivity extends Activity
{
TextView hello1;
Button display;
@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
hello1 = (TextView) findViewById(R.id.txthello1);
hello1.setText("HELLO MITS MCA");
display = (Button) findViewById(R.id.btndisp);
display.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View arg0)
{
Toast t = Toast.makeText(getApplicationContext(), "MCA", 2000);
t.show();
}
});
}
}
```

17. From the Menu bar Choose Window and Select Android Virtual Device Manager.

18. Choose New and Give a name for the Virtual Device (Emulator)

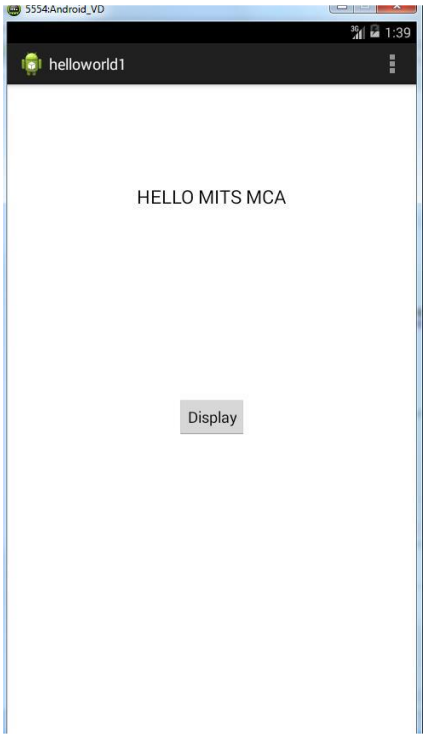
19. Choose the Device Name as WVGA 5.1

20. Click ok.

21. Goto explorer window and right click on the project hello and choose Run As -> Android application.

22. The Emulator will give the output as per the requirement

Output:



Result:

Android application to display the given text in a TextView control and a Toast has been created successfully**.**

Exercise 2:

Aim:

To create an Android application that uses the Internal Keyboard to take input, using Input Method Framework.

Procedure:

1. Open adk Bundle software.
2. Open Eclipse Folder and open eclipse .exe file
3. From File Menu, Choose New -> Android Application Project
4. Give keyboard as the project name
5. Click Next and Choose the Blank Activity and Click Finish
6. In the explorer window choose the res folder and choose layout
7. Double click Layout and open the activity_main.xml file by double clicking the file
8. From the bottom Tab, Choose Graphical Layout.
9. Place a TextView control, an EditText Control and a Button control.
10. Right click on TextView Control click on EditText and change the Text as "Press any key on the keyboard" and give new R.string name as input.
11. Right click on button control and click on EditText and change the Text as "Display the keyboard" and give the new R. string name as display
12. Right click on EditText control and click on Edit Id and change the id as "txtinput" and give the new R.string name as dispinput.
13. Write the following coding in the MainActivity.java file

```
public class MainActivity extends Activity {
Button btn1;
EditText et1;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
btn1 = (Button) findViewById(R.id.button1);
et1 = (EditText) findViewById(R.id.txtinput);
btn1.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View arg0)
{
// TODO Auto-generated method stub
InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
imm.toggleSoftInput(InputMethodManager.SHOW_FORCED, 0); }
});
}}}
```

14. From the Menu bar Choose Window and Select Android Virtual Device Manager.

15. Choose New and Give a name for the Virtual Device (Emulator)

16. Choose the Device Name as WVGA 5.1

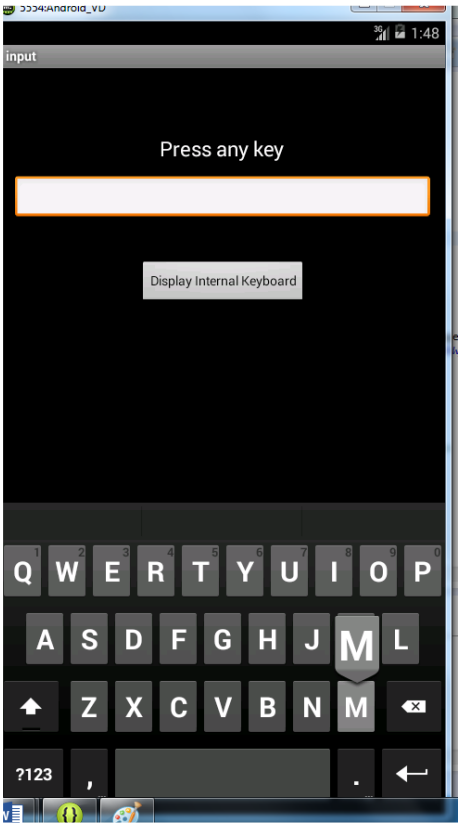
17. Uncheck the Hardware Keyboard Present checkbox

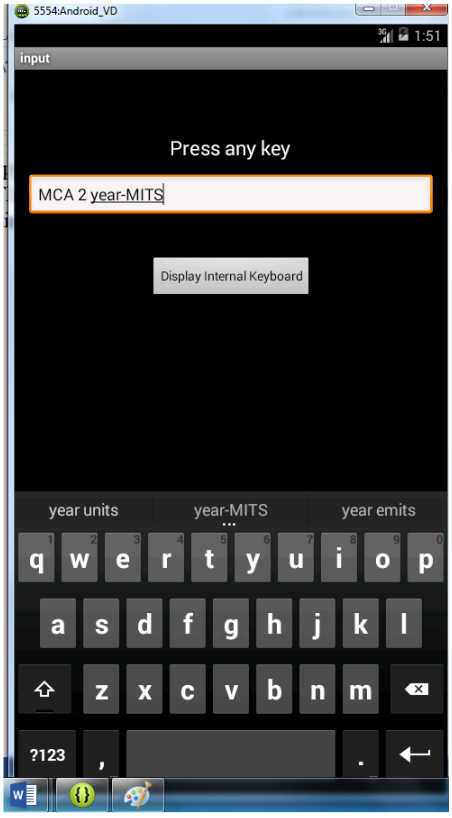
18. Click ok.

19. Goto explorer window and right click on the project keyboard and choose Run As -> Android application.

20. The keyboard will appear and the input can be given through the internal keyboard which will be displayed in the emulator

Output:





Result:

The keyboard has been created successfully to get the input using Input Method Manager

Exercise 3. i.

Aim:

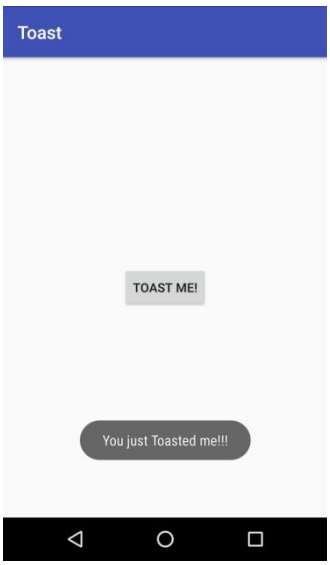
To develop an android application to display a text in the toast

Procedure:

1. Open Android Eclipse and create a new Android application and name it as **"Toast"**..
2. Open an Empty activity and name it as **MainActivity**.
3. Drag and drop a button in the interface.
4. Give String Name and R.String name as toast.
5. Write the following code in the Mainactivity.java file

```
public class MainActivity extends AppCompatActivity {
    Button b1;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      b1 = (Button) findViewById(R.id.toast_button);
       b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Toast.makeText(MainActivity.this,"You just Toasted
me!!!",Toast.LENGTH_LONG).show();
            /*Another way to display a Toast message
               Toast t=Toast.makeText(MainActivity.this,"You just Toasted
me!!!",Toast.LENGTH_LONG);
               t.show();
            /
         }
      });
   }

}
```

Output:



Result:

Android application to display toast has been created successfully

Exercise 3. ii

Aim:

To develop an android application to display the text message in the toast.

Procedure:

1. Open android eclipse.
2. Place necessary controls in the interface
3. Write the following code in the MainActivity.java file

```
package com.example.admin.myapplication;
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 {
   EditText enterValuesEditText;
   Button   clickHereButton;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      enterValuesEditText =(EditText)findViewById(R.id.enterValueEditText);
      clickHereButton =(Button)findViewById(R.id.clickHereButton);
      clickHereButton.setOnClickListener(new View.OnClickListener() {
@Override
 public void onClick(View view) {
Toast.makeText(MainActivity.this,enterValuesEditText.getText().toString(),Toast
.LENGTH_LONG).show();
         }
      });
   }
}
```

Output:

Result:

Android application to display the text message in the toast has been executed successfully’

Exercise 4:

Aim:

To Develop an application to perform 5 arithmetic operations: Addition, Subtraction, Multiplication, Division and Modulo operation with necessary user interface creation

Procedure:

1. Open adk Bundle software.
2. Open Eclipse Folder and open eclipse .exe file
3. From File Menu, Choose New -> Android Application Project
4. Give checkbox as the project name
5. Click Next and Choose the Blank Activity and Click Finish
6. In the explorer window choose the res folder and choose layout
7. Double click Layout and open the activity_main.xml file by double clicking the file
8. From the bottom Tab, Choose Graphical Layout
9. Place three TextView controls, three EditText controls and four Buttons
10. Right click on each TextView control and choose EditText option and give the text as First Number, Second Number and Result respectively
11. Right click on the Edit Text controls and give new R.String as add, sub and result respectively
12. Right click on the button controls and choose EditText option and give names as Add, Sub, Mul and Div for those buttons
13. Write the following coding in the MainActivity.java file

```
public class MainActivity extends Activity {
EditText et1, et2, et3;
Button b1, b2, b3, b4;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
et1 = (EditText) findViewById(R.id.editText1);
et2 = (EditText) findViewById(R.id.editText2);
et3 = (EditText)findViewById(R.id.editText3);
b1 = (Button) findViewById(R.id.button1);
b2 = (Button) findViewById(R.id.button2);
b3 = (Button) findViewById(R.id.button3);
b4 = (Button) findViewById(R.id.button4);
b1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
String f = et1.getText().toString();
int i = Integer.parseInt(f);
String s = et2.getText().toString();
int j = Integer.parseInt(s);
Integer result = i + j;
String res = result.toString();
et3.setText(res);
```

```
}
});
b2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
String f = et1.getText().toString();
int i = Integer.parseInt(f);
String s = et2.getText().toString();
int j = Integer.parseInt(s);
Integer result = i - j;
String res = result.toString();
et3.setText(res);
}
});
b3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
String f = et1.getText().toString();
int i = Integer.parseInt(f);
String s = et2.getText().toString();
int j = Integer.parseInt(s);
Integer result = i * j;
String res = result.toString();
et3.setText(res);
}
});
b4.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
String f = et1.getText().toString();
int i = Integer.parseInt(f);
String s = et2.getText().toString();
int j = Integer.parseInt(s);
Integer result = i / j;
String res = result.toString();
et3.setText(res);
}
});
}
```

14. From the Menu bar Choose Window and Select Android Virtual Device Manager.

15. Choose New and Give a name for the Virtual Device (Emulator)

16. Choose the Device Name as WVGA 5.1

17. Click ok.

18. Goto explorer window and right click on the project checkbox and choose Run As -> Android application.

19. The android application will perform the arithmetic operation in the android virtual device (Emulator)

Output:

Addition Operation

Subtraction Operation

Multiplication Operation

Division Operation

Result:

The android application has been created to perform the arithmetic operations successfully

Exercise: 5

Aim:

To create an android application that performs the operations of a simple calculator

Procedure:

1. Open adk Bundle software.
2. Open Eclipse Folder and open eclipse .exe file
3. From File Menu, Choose New -> Android Application Project
4. Give checkbox as the project name
5. Click Next and Choose the Blank Activity and Click Finish
6. In the explorer window choose the res folder and choose layout
7. Double click Layout and open the activity_main.xml file by double clicking the file
8. From the bottom Tab, Choose Graphical Layout
9. Place TextView controls and EditText controls and 17 Buttons
10. Right click on the TextView controls and give new R.String as tv1
11. Right click on the EditText controls and give new R.String as ed1
12. Right click on the button controls and choose EditText option and give names as Add, Sub, Mul,
Div,clear,digit0, digit1, digit2, digit3, digit4, digit5, digit6, digit7, digit8, digit9 for those buttons
13. Write the following coding in the MainActivity.java file

```
public class MainActivity extends Activity {
Button
clear,div,mul,sub,dot,equal,add,btn1,btn2,btn3,btn4,btn5,btn6,btn7,btn8,btn9,btn0;
float Res1,Res2;
boolean Add,Sub,Mul,Div;
EditText ed1;
TextView tv1;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
ed1 = (EditText)findViewById(R.id.display);
tv1 = (TextView)findViewById(R.id.tv1);
btn0 = (Button)findViewById(R.id.digit0);
btn1 = (Button)findViewById(R.id.digit1);
btn2 = (Button)findViewById(R.id.digit2);
btn3 = (Button)findViewById(R.id.digit3);
btn4 = (Button)findViewById(R.id.digit4);
btn5 = (Button)findViewById(R.id.digit5);
btn6 = (Button)findViewById(R.id.digit6);
btn7 = (Button)findViewById(R.id.digit7);
btn8 = (Button)findViewById(R.id.digit8);
btn9 = (Button)findViewById(R.id.digit9);
clear = (Button)findViewById(R.id.clear);
div = (Button)findViewById(R.id.div);
mul = (Button)findViewById(R.id.mul);
sub = (Button)findViewById(R.id.minus);
dot = (Button)findViewById(R.id.dot);
equal = (Button)findViewById(R.id.equal);
add = (Button)findViewById(R.id.plus);
btn1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
```

```
ed1.setText(ed1.getText()+"1");
}
});
btn2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"2");
}
});
btn3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"3");
}
});
btn4.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"4");
}
});
btn5.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"5");
}
});
btn6.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"6");
}
});
btn7.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"7");
}
});
btn8.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"8");
}
});
btn9.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+"9");
}
});
btn0.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
```

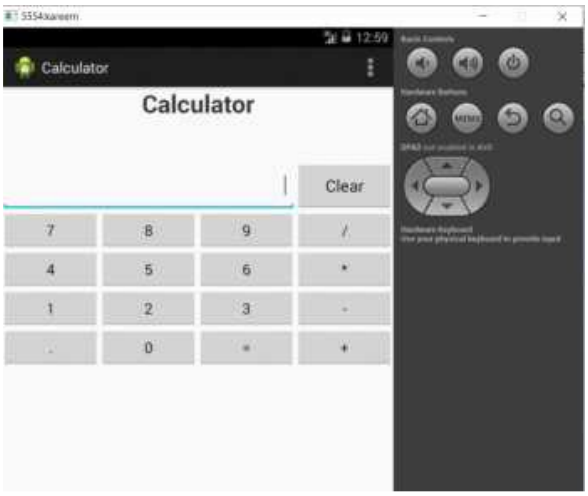
```
ed1.setText(ed1.getText()+"0");
}
});
dot.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText(ed1.getText()+".");
}
});
add.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
if(ed1==null){
ed1.setText("");
} else {
Res1=Float.parseFloat(ed1.getText()+"");
Add=true;
tv1.setText(ed1.getText());
ed1.setText(null);
}
}
});
sub.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
if(ed1==null){
ed1.setText("");
} else {
Res1=Float.parseFloat(ed1.getText()+"");
Sub=true;
tv1.setText(ed1.getText());
ed1.setText(null);
}
}
});
mul.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
if(ed1==null){
ed1.setText("");
} else {
Res1=Float.parseFloat(ed1.getText()+"");
Mul=true;
tv1.setText(ed1.getText());
ed1.setText(null);
}
}
});
div.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
if(ed1==null){
ed1.setText("");
} else {
```

```
Res1=Float.parseFloat(ed1.getText()+"");
Div=true;
tv1.setText(ed1.getText());
ed1.setText(null);
}
}
});
equal.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
Res2=Float.parseFloat(ed1.getText()+"");
if(Add==true){
tv1.setText(tv1.getText()+"+"+ed1.getText());
ed1.setText(Res1+Res2+"");
tv1.setText(tv1.getText()+"="+ed1.getText());
Add=false;
}
if(Sub==true){
tv1.setText(tv1.getText()+"-"+ed1.getText());
ed1.setText(Res1-Res2+"");
tv1.setText(tv1.getText()+"="+ed1.getText());
Sub=false;
}
if(Mul==true){
tv1.setText(tv1.getText()+"*"+ed1.getText());
ed1.setText(Res1*Res2+"");
tv1.setText(tv1.getText()+"="+ed1.getText());
Mul=false;
}
if(Div==true){
tv1.setText(tv1.getText()+"/"+ed1.getText());
ed1.setText(Res1/Res2+"");
tv1.setText(tv1.getText()+"="+ed1.getText());
Div=false;
}
}
});
clear.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
ed1.setText("");
tv1.setText("");
}
});
}
@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;
}
}
```

14. From the Menu bar Choose Window and Select Android Virtual Device Manager.

15. Choose New and Give a name for the Virtual Device (Emulator)
16. Choose the Device Name as WVGA 5.1
17. Click ok.
18. Goto explorer window and right click on the project checkbox and choose Run As -> Android
application.
19. The android application will perform the arithmetic operation in the android virtual device

Output:



Result:

Android application for a calculator has been created successfully'

Exercise 6:

Aim:

To Create an android UI that consists of Different Departments of a company namely Production, Finance, Marketing and HR. If the user clicks on any department it should show details of that department. Use indents

Procedure:

1. Open Eclipse.
2. Place 4 buttons in the interface.
3. Name them as HR, Marketing, Finance and Accounts.
4. Create three more activities and name them am Second, Third and Fourth Activity respectively
5. In these activities place text view controls only to display the details of the department when the button is clicked
6. Type the following coding in the MainActivity.java file

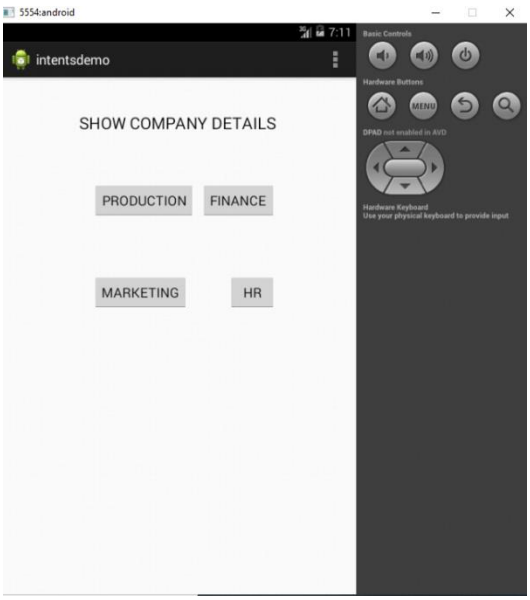
```
package com.example.intends_demo;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
    Button b1,b2,b3,b4;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      b1 = (Button)findViewById(R.id.button1);
      b2 = (Button)findViewById(R.id.button2);
      b3 = (Button)findViewById(R.id.button3);
      b4 = (Button)findViewById(R.id.button4);
      b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Intent i = new Intent(MainActivity.this, Prod.class);
            startActivity(i);
            }
            });
       b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                  // TODO Auto-generated method stub
                  Intent i1 = new Intent(MainActivity.this, Fin.class);
                  startActivity(i1);
                  }
            });
      b3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                  // TODO Auto-generated method stub
            Intent i2 = new Intent(MainActivity.this, Mar.class);
```

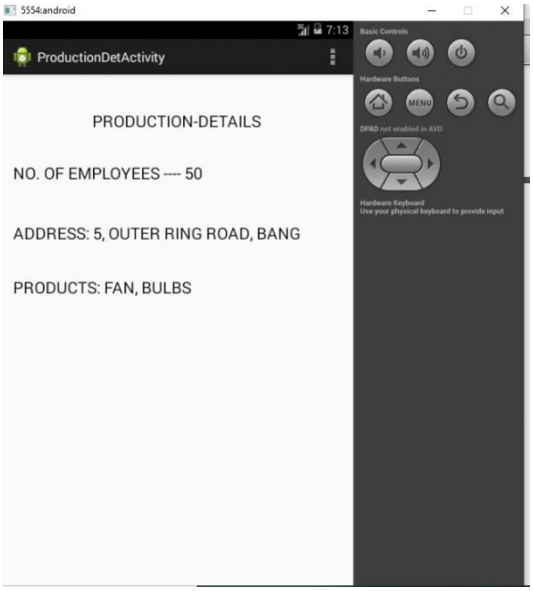
```
            startActivity(i2);
            }
    });
      }
   @Override
    public boolean onCreateOptionsMenu(Menu menu) {
       // Inflate the menu; this adds items to the action bar if it is present.
       getMenuInflater().inflate(R.menu.main, menu);
       return true;
    }
}
```

7. Type the following in all the other activities

```
package com.example.intends_demo;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class Fin extends Activity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_fin);
        }
        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
                // Inflate the menu; this adds items to the action bar if it is present.
                getMenuInflater().inflate(R.menu.fin, menu);
                return true;
        }
}
```

Output:





MAIN SCREEN AFTER CLICKING PRODUCTION BUTTON

Result:

Android application to demonstrate the intents working methodology has been created successfully''

Exercise 7:

Aim:

To design an android application to display a list of items on the android screen. If the user clicks any one of the list items a dialogue box should show that the user has clicked that particular item (Use array adopters)

Procedure:

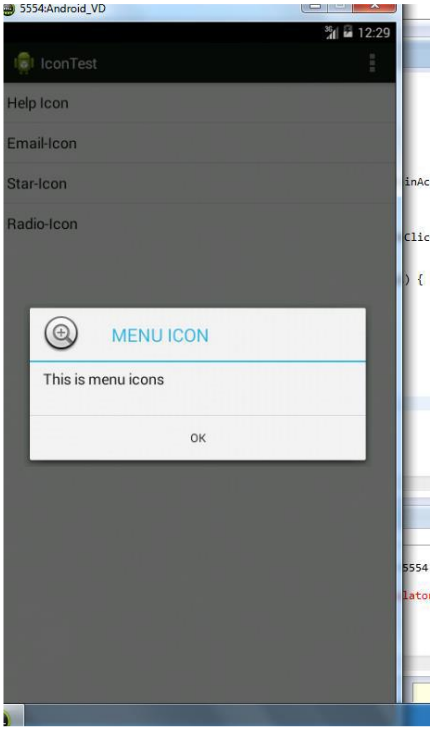
1. Open adk Bundle software.
2. Open Eclipse Folder and open eclipse .exe file
3. From File Menu, Choose New -> Android Application Project
4. Give list_icon as the project name
5. Click Next and Choose the Blank Activity and Click Finish
6. In the explorer window choose the res folder and choose layout
7. Double click Layout and open the activity_main.xml file by double clicking the file
8. From the bottom Tab, Choose Graphical Layout
9. Place a ListView control into the layout by selecting the Composite option from the widget button
10. Right click on ListView and give R.new String as list
11. Write the following coding in the MainActivity.java file

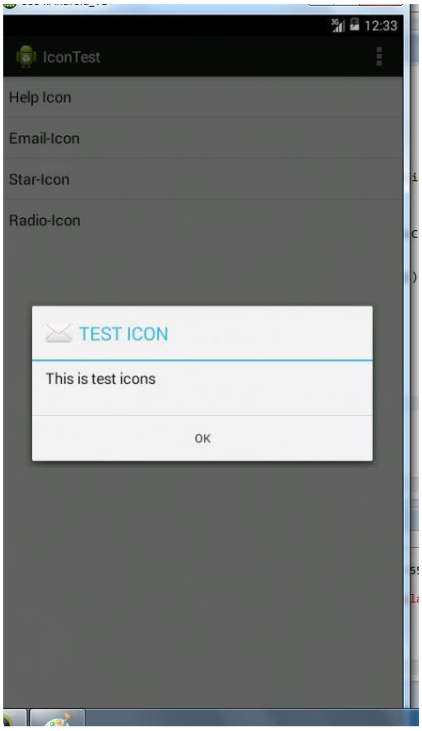
```
public class MainActivity extends Activity {
ListView icons;
String iconlist[]=new String[]{"Help Icon","Email-Icon","Star-Icon","Radio-Icon"};
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
icons=(ListView) findViewById(R.id.listView1);
ArrayAdapter embed=new ArrayAdapter<String>(MainActivity.this,
android.R.layout.simple_list_item_1, iconlist);
icons.setAdapter(embed);
icons.setOnItemClickListener(new OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int pos,
long arg3) {
// TODO Auto-generated method stub
switch(pos)
{
case 0:
        AlertDialog.Builder dialog=new AlertDialog.Builder(MainActivity.this);
        dialog.setTitle("MENU ICON");
        dialog.setMessage("This is menu icons");
        dialog.setIcon(android.R.drawable.btn_plus);
        dialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {
```

```
        @Override
        public void onClick(DialogInterface arg0, int arg1) {
        // TODO Auto-generated method stub
        arg0.cancel(); }
        });
        Dialog d=dialog.create();
        d.show();
        break;
case 1:
AlertDialog.Builder dialog1=new AlertDialog.Builder(MainActivity.this);
        dialog1.setTitle("TEST ICON");
        dialog1.setMessage("This is test icons");
        dialog1.setIcon(android.R.drawable.ic_dialog_email);
        dialog1.setPositiveButton("OK", new DialogInterface.OnClickListener()
        {
        @Override
        public void onClick(DialogInterface arg0, int arg1) {
        // TODO Auto-generated method stub
        arg0.cancel();
        }
        });
        Dialog d1=dialog1.create();
        d1.show();
        break;
case 2:
        AlertDialog.Builder dialog2=new AlertDialog.Builder(MainActivity.this);
        dialog2.setTitle("TEST ICON");
        dialog2.setMessage("This is test icons");
        dialog2.setIcon(android.R.drawable.btn_star);
        dialog2.setPositiveButton("OK", new DialogInterface.OnClickListener()
        {
        @Override
        public void onClick(DialogInterface arg0, int arg1) {
        // TODO Auto-generated method stub
        arg0.cancel();
        }
        });
        Dialog d2=dialog2.create();
        d2.show();
        break;
case 3:
        AlertDialog.Builder dialog3=new AlertDialog.Builder(MainActivity.this);
        dialog3.setTitle("TEST ICON");
        dialog3.setMessage("This is test icons");
        dialog3.setIcon(android.R.drawable.btn_radio);
        dialog3.setPositiveButton("OK", new DialogInterface.OnClickListener()
        {
        @Override
        public void onClick(DialogInterface arg0, int arg1) {
        // TODO Auto-generated method stub
        arg0.cancel();
        }
        });
        Dialog d3=dialog3.create();  d3.show();  break; } } }); }
```

12. From the Menu bar Choose Window and Select Android Virtual Device Manager.
13. Choose New and Give a name for the Virtual Device (Emulator)
14. Choose the Device Name as WVGA 5.1
15. Click ok.
16. Goto explorer window and right click on the project list_icon and choose Run As -> Android application.
17. Proper icons will be displayed while clicking the list items

Output:





Result:

Thus the android application for selection list and the Icons is created successfully in Android

Exercise – 8:

Aim:

To develop an android application to show some categories such as education, entertainment, health, provisions etc., If the user clicks on any one of the items it should show the sub categories of the category and if is again clicked it should the details of those items. (Use indents and lists)

Procedure:

1. Open adk Bundle software.
2. Open Eclipse Folder and open eclipse .exe file
3. From File Menu, Choose New -> Android Application Project
4. Give list_icon as the project name
5. Click Next and Choose the Blank Activity and Click Finish
6. In the explorer window choose the res folder and choose layout
7. Double click Layout and open the activity_main.xml file by double clicking the file
8. From the bottom Tab, Choose Graphical Layout
9. Click on Composite in the Widget
10. From the controls drag listview control and drag it to the layout
11. Right click on the listview and give R. new String name as listview1
12. From the res folder choose on the menu folder
13. Right Click on the menu folder and choose New ->Android File
14. Give a name for the menu
15. Click Add
16. From the dialogue select browse and give string name and R String name as save, edit, delete and update one by one.
17. Write the following coding in MainActivity.java file

```
public class MainActivity extends Activity {
ListView lv;
String data[]=new
=String[]{"Chennai","Trichy","Madurai","Karur","Covai","Salem"};
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
lv=(ListView)findViewById(R.id.listView1);
ArrayAdapter<String>
adap=new ArrayAdapter<String>(getApplicationContext(),
android.R.layout.simple_list_item_1,data);
lv.setAdapter(adap);
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
// TODO Auto-generated method stub
// TODO Auto-generated method stub
switch(item.getItemId())
{
case R.id.save:
        Toast.makeText(getApplicationContext(), "You clicked Save MenuItem.",
        Toast.LENGTH_LONG).show();
        break;
case R.id.edit:
        Toast.makeText(getApplicationContext(), "You clicked Edit MenuItem.",
        Toast.LENGTH_LONG).show(); break;
```

```
case R.id.delete:
        Toast.makeText(getApplicationContext(), "You clicked Delete MenuItem.",
        Toast.LENGTH_LONG).show();
        break;
```

```
        case R.id.update:
                Toast.makeText(getApplicationContext(), "You clicked Update MenuItem.",
                Toast.LENGTH_LONG).show();
                break;
                }
                return false; }
@Override
        public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.option_menu, menu);
        return true;
        }
        }
```

18. Right click on the OptionsMenu application and click RunAs ->Android application
19. The emulator will show the list of items.
20. Click on Menu button in the right side of the Emulator
21. A menu will appear at the bottom of the emulator
22. Click on each button and perform the necessary operation such as Save, Edit, Delete or Update

Output:

Result:

Android application to display list item has been created successfully

Exercise 9.i.

Aim:

To Design an android application to create a service that shows the service is running in the background in the form of a toast

Procedure:

1. Open Ecliplse Exe.
2. Choose New project
3. Choose Android application project
4. Give the project name as ***service***
5. Click next and choose blank activity.
6. In the Graphical layout window keep two button controls and give the String name as 'START SERVICE' and STOP SERVICE.
7. Give R.String name as start and stop
8. Open layout.xml file and type the following in the two Button classes.
   android:onClick = "startMethod" (this is for start button class)
   android:onClick = "stopMethod" (this is for stop button class)
9. Goto Explorer window and choose src folder and choose com.service
10. Choose New -> Class and give a class name in the dialogue box as MyFirstService
11. Type the following in the class file.

```
public class MyFirstService extends service
        {
                @Override
                publicIBinderonBind (Intent intent) {
                // TODO Auto generated method stud
                return null;
                }
                @Override
                public void onCreate() {
                        // TODO Auto generated method stud
                    super.onCreate();
                    Toast.maketext(this, "Service has been created",
                    Toast.LENGTH_LONG).show();
                    }
                    @Override
                    public void onStartCommand(Intent intent, int flags, intstartid)
{
                        // TODO Auto generated method stud
                    Toast.maketext(this, "Service has been started",
                    Toast.LENGTH_LONG).show();
                    returnsuper.onStartCommand(intent, flags, startid);
                    }
                    @Override
                    public void onDestroy() {
                        // TODO Auto generated method stud
                    Toast.maketext(this, "Service is stopped",
                    Toast.LENGTH_LONG).show();
                    super.onDestroy();
```

}

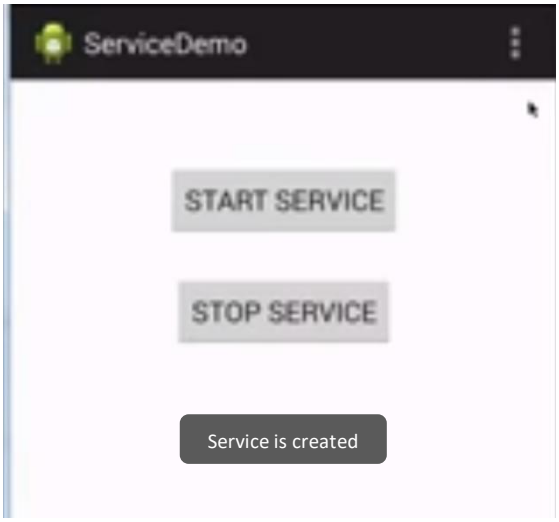
12. Open Manifest file from the explorer window and choose manifest.xml file from the bottom tab in the dialogue box shown.
13. In the file type the following after the </activity>

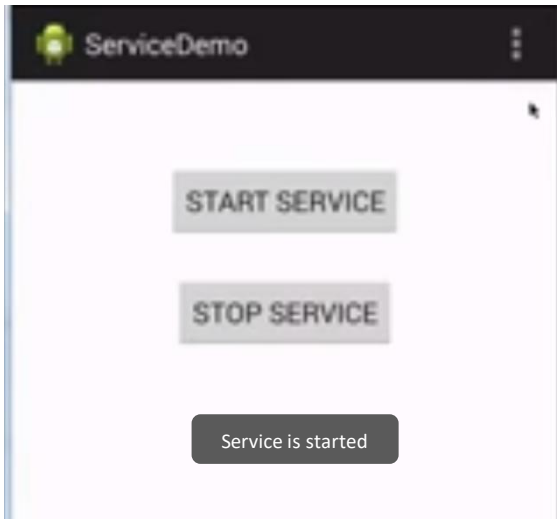
```
</activity>
<service
        android:name = ".service"
        android:exported = "false"
/>
```

14. Goto Mainactivity.java file and type the following code

```
Public void startMethod(View v)
        {
        Intent i= new Intent(this);
        startService(i);
        }
Public void stopMethod(View v)
        {
        Intent i = new Intent(this);
        stopService(i);
        }
```

Output:





Result:

Android application to demonstrate the working methodology of a services is created successfully

Exercise 9. ii.

Aim:

To create an android application to show the working methodology of an alarm clock in android.

Procedure

1. Open adk bundle
2. Place an edit text control and a button control on the activity after creating a blank activity. click save all
3. Open Mainactivity.java and write the following code.

```
package example.alarmmanager;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
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 {
```

4. Create one more class file for Broadcast receiver and write the following code in the Broadcastreceiver.class file

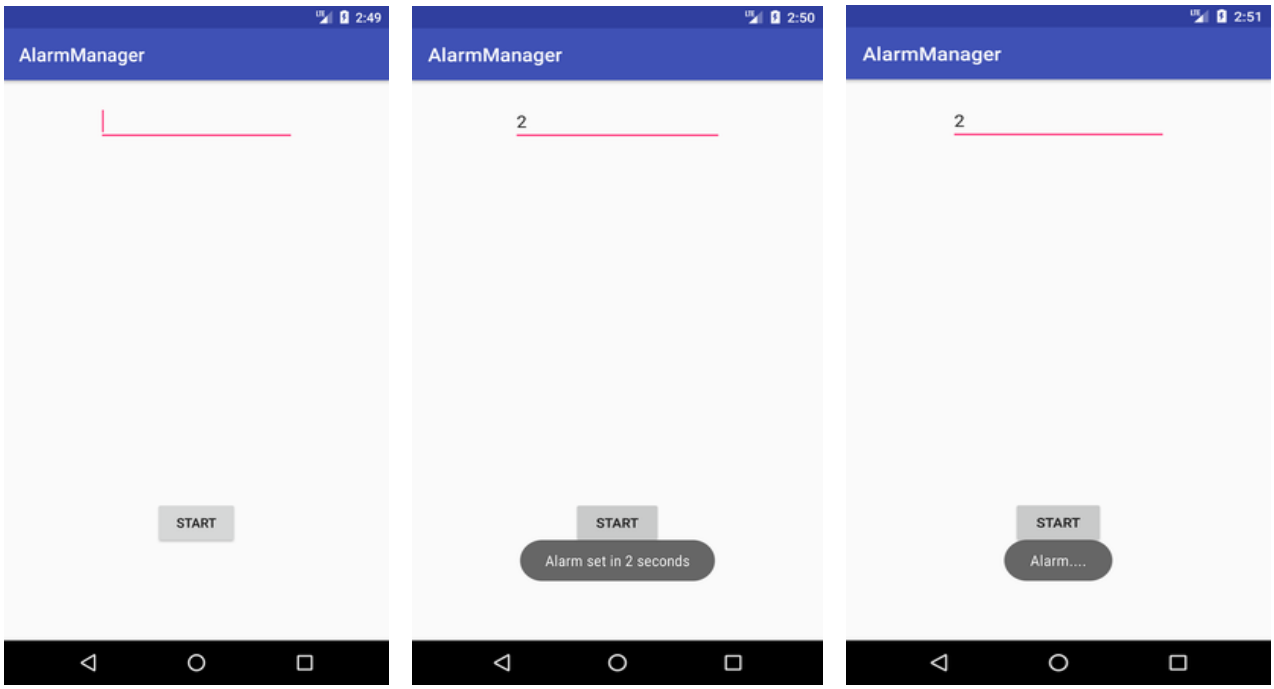
```
package example.alarmmanager;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.widget.Toast;
public class MyBroadcastReceiver extends BroadcastReceiver {
   MediaPlayer mp;
   @Override
   public void onReceive(Context context, Intent intent)
 {
      //Todo the method stud
    mp=MediaPlayer.create(context, R.raw.alarm);
    mp.start();
    Toast.makeText(context, "Alarm....", Toast.LENGTH_LONG).show();
   }
}
```

5. **Open android Manifest xml file and type the following code**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="example.alarmmanager">
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
```

```
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
         <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <receiver
          android:name="MyBroadcastReceiver" >
    </receiver>
  </application>
</manifest>
```

Output:



Result:

Android application for alarm services is created and executed successfully.

Note:

To start an Alarm Manager you need to first get the instance from the System. Then pass the PendingIntent which would get executed at a future time that you specify.

```
AlarmManager manager = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
Intent alarmIntent =
          new Intent(context, MyAlarmReceiver.class);
          PendingIntent pendingIntent = PendingIntent.getBroadcast(context,
          0, alarmIntent, 0);
int interval = 8000;
manager.setInexactRepeating(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis(), interval, pendingIntent);
```

Exercse 10:

Aim:

To develop an android application to demonstrate the database connectivity with the SQLite database to post and retrieve data through the User Interface(Example: Student mark list processing, Email Registration and Login, Products and sales)

Procedure:

1. Open adk Bundle software.
2. Open Eclipse Folder and open eclipse .exe file
3. From File Menu, Choose New -> Android Application Project
4. Give Database as the project name
5. Click Next and Choose the Blank Activity and Click Finish
6. In the explorer window choose the res folder and choose layout
7. Double click Layout and open the activity_registration.xml file
8. From the bottom Tab, Choose Graphical Layout.
9. From the pallete in the left side of the graphical window drag a EditText control and place it in the activity window.
10. Similarly, drag a button control from the pallete and place it in the activity window.
11. Right Click on the button and choose Edit Text and click New String
12. Give 'Register' as String name and register as R.New String name
13. Right Click on the EditText control and give R.New String name as name, email, password.
14. Right Click on the src folder in the explorer window and choose the RegistrationActivity.java file
15. Double click the **RegistrationActivity.java** file and write the following in the file in the public class

```
@SuppressLint("NewApi") public class RegistrationActivity extends Activity
{
EditText name, email, password;
String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\\.+[a-z]+";
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_registration);
name = (EditText) findViewById(R.id.etname);
email = (EditText) findViewById(R.id.etusername);
password = (EditText) findViewById(R.id.etpass);
}
public void Reg(View view){
if(!validate()){
Toast.makeText(this, "Signup has Failed",
Toast.LENGTH_SHORT).show();
} else {
String str_name = name.getText().toString().trim();
String str_email = email.getText().toString().trim();
String str_password = password.getText().toString().trim();
String type = "register";
BackgroundWorker backgroundWorker = new BackgroundWorker(this);
backgroundWorker.execute(type, str_name, str_email,
str_password);
}
}
```

```
@SuppressLint("NewApi") private boolean validate() {
boolean valid = true;
        if(name.getText().toString().isEmpty() &&
        email.getText().toString().isEmpty() &&
        password.getText().toString().isEmpty()){
        name.setError("Please enter email");
        email.setError("Plese enter email");
        password.setError("Please enter confirm password");
        valid = false;
        }
        if(!email.getText().toString().matches(emailPattern)){
        email.setError("Enter correct email eg: kareem@gmail.com");
        valid = false;
        }
        if(name.getText().toString().length()>3 &&
        (name.getText().toString().length()<32)){
        } else {
        name.setError("min 3 and max 32");
        valid = false;
        }
        if(password.getText().toString().length()>3 &&
        (password.getText().toString().length()<8)){
        } else {
        password.setError("min 3 and max 8");
        valid = false;
        }
        return valid;
        }
@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is
present.
getMenuInflater().inflate(R.menu.registration, menu);
return true;
}
}
```

16. Create one more class called LoginActivity.java and type the following code

```
@SuppressLint("NewApi") public class LoginActivity extends Activity {
EditText emailEt,passwordEt;
TextView reg_link;
Button login;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
emailEt = (EditText) findViewById(R.id.login_username);
passwordEt = (EditText) findViewById(R.id.login_pass);
reg_link = (TextView) findViewById(R.id.reg_link);
login = (Button) findViewById(R.id.btnlogin);
reg_link.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
startActivity(new Intent(getApplication(),
RegistrationActivity.class));
```

```
}
});
login.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
// TODO Auto-generated method stub
login();
}
});
}
public void login() {
        if(!validate()){
        Toast.makeText(this, "Login has Failed",
        Toast.LENGTH_SHORT).show();
        } else {
        String email = emailEt.getText().toString();
        String password = passwordEt.getText().toString();
        String type = "login";
        BackgroundWorker backgroundWorker = new BackgroundWorker(this);
        backgroundWorker.execute(type, email, password);
        }
        }
@SuppressLint("NewApi") private boolean validate() {
boolean valid = true;
        if(emailEt.getText().toString().isEmpty()
        && passwordEt.getText().toString().isEmpty()){
        emailEt.setError("Please enter email");
        passwordEt.setError("Please enter password");
        valid = false;
        }
        return valid;
        }
@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;
}
}
```

17. .Create java file **BackgroundWorker.java**

```
public class BackgroundWorker extends AsyncTask<String, Void, String>{
Context context;
AlertDialog alertDialog;
BackgroundWorker (Context ctx) {
context = ctx;
}
@Override
protected String doInBackground(String... params) {
String type = params[0];
String register_url = "https://ropiestfinishes.
000webhostapp.com/Database%20eclipse/register.php";
String login_url = "https://ropiestfinishes.
000webhostapp.com/Database%20eclipse/Login.php";
```

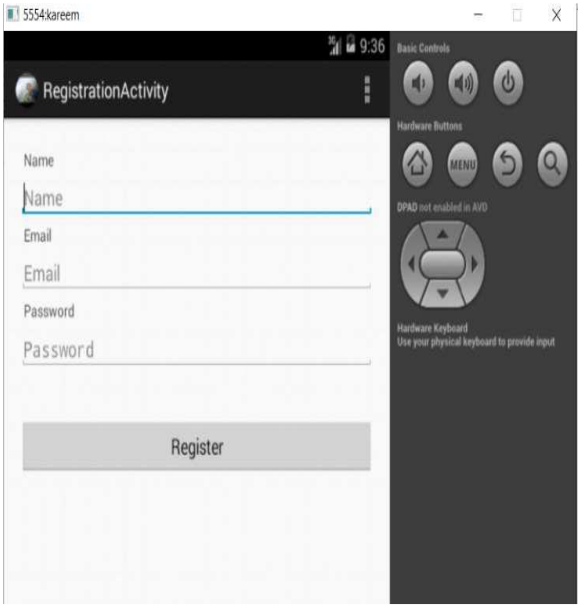
```
if(type.equals("login")) {
try {
String email = params[1];
String password = params[2];
URL url = new URL(login_url);
HttpURLConnection httpURLConnection =
(HttpURLConnection)url.openConnection();
httpURLConnection.setRequestMethod("POST");
httpURLConnection.setDoOutput(true);
httpURLConnection.setDoInput(true);
OutputStream outputStream =
httpURLConnection.getOutputStream();
BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
String post_data = URLEncoder.encode("email","UTF-
8")+"="+URLEncoder.encode(email,"UTF-8")+"&"
+URLEncoder.encode("password","UTF-
8")+"="+URLEncoder.encode(password,"UTF-8");
bufferedWriter.write(post_data);
bufferedWriter.flush();
bufferedWriter.close();
outputStream.close();
InputStream inputStream =
httpURLConnection.getInputStream();
BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
String result="";
String line="";
while((line = bufferedReader.readLine())!= null) {
result += line;
}
bufferedReader.close();
inputStream.close();
httpURLConnection.disconnect();
return result;
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
} else if(type.equals("register")) {
try {
String name = params[1];
String age = params[2];
String email = params[3];
String password = params[4];
URL url = new URL(register_url);
HttpURLConnection httpURLConnection =
(HttpURLConnection)url.openConnection();
httpURLConnection.setRequestMethod("POST");
httpURLConnection.setDoOutput(true);
httpURLConnection.setDoInput(true);
OutputStream outputStream =
httpURLConnection.getOutputStream();
```

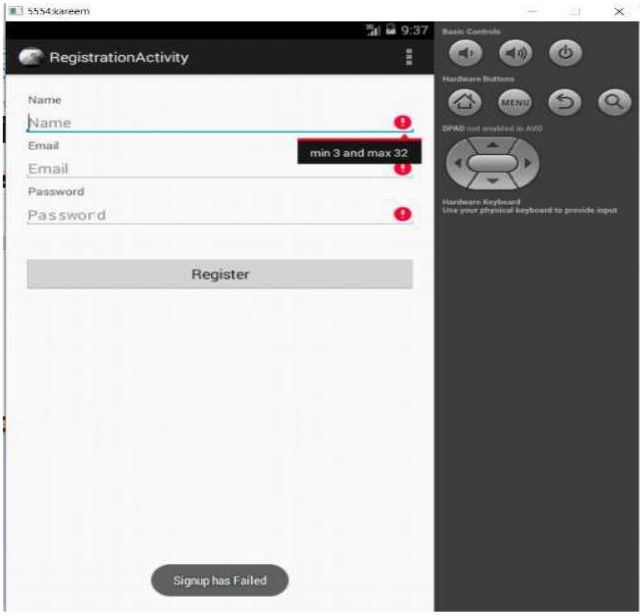
```
BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
String post_data = URLEncoder.encode("name","UTF-
8")+"="+URLEncoder.encode(name,"UTF-8")+"&"
+URLEncoder.encode("email","UTF-
8")+"="+URLEncoder.encode(email,"UTF-8")+"&"
+URLEncoder.encode("password","UTF-
8")+"="+URLEncoder.encode(password,"UTF-8");
bufferedWriter.write(post_data);
bufferedWriter.flush();
bufferedWriter.close();
outputStream.close();
InputStream inputStream =
httpURLConnection.getInputStream();
BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
String result="";
String line="";
while((line = bufferedReader.readLine())!= null) {
result += line;
}
bufferedReader.close();
inputStream.close();
httpURLConnection.disconnect();
return result;
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}
return null;
}
@Override
protected void onPreExecute() {
alertDialog = new AlertDialog.Builder(context).create();
alertDialog.setTitle("Login Status");
}
@Override
protected void onPostExecute(String result) {
alertDialog.setMessage(result);
alertDialog.show();
}
@Override
protected void onProgressUpdate(Void... values) {
super.onProgressUpdate(values);
}
}
```

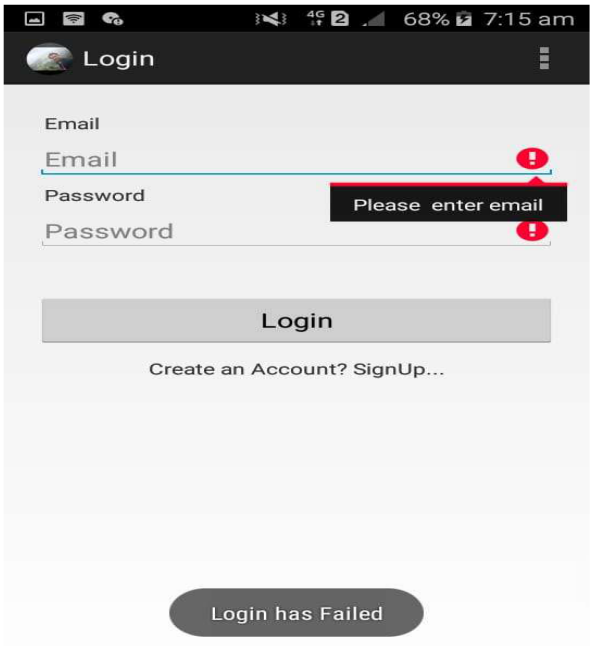
17. Open the AndroidManifest.xml file and add the permission
    <uses-permission android:name=”android.permission.INTERNET”/>
18. From the Menu bar Choose Window and Select Android Virtual Device Manager.
19. Choose New and Give a name for the Virtual Device (Emulator)
20. Choose the Device Name as WVGA 5.1 20. Click ok.

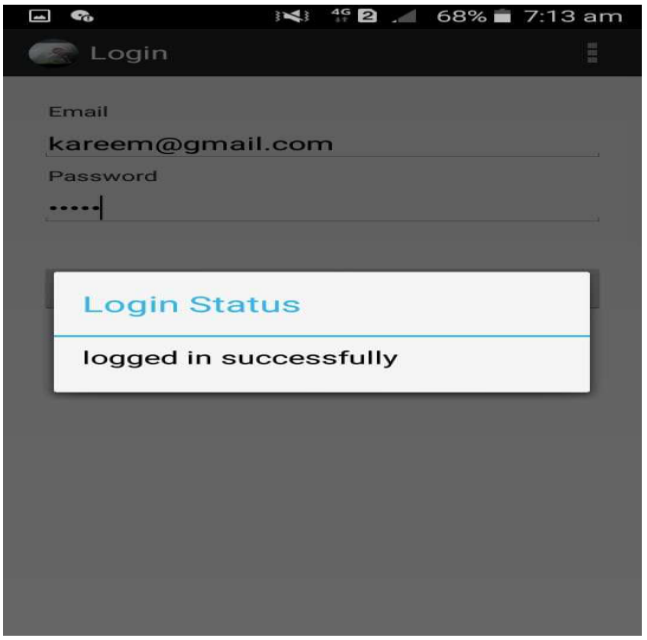
21. Goto explorer window and right click on the project hello and choose Run As -> Android application.
22. The Emulator will give the output as per the requirement

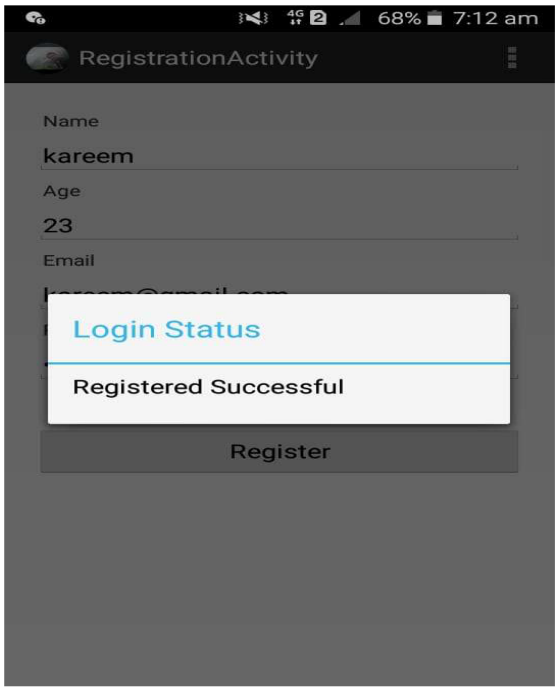
Output:

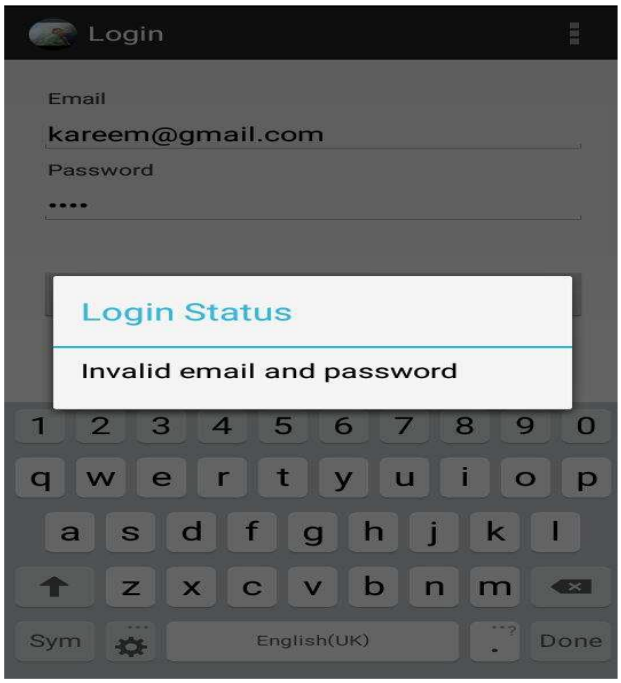












Result:

Android application to demonstrate the working procedure of database connectivity for registering a user has done successfully.

Exercise 11:

a. Aim:

To demonstrate the usage of Sensors in android by developing proper application

Procedure:

1. Open adk bundle and open eclipse
2. Place a textview control and give R.string name
3. Type the following code in the MainActivity java file

```
package com.example.sensorsimple;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import android.hardware.SensorManager;
import android.hardware.SensorEventListener;
import android.hardware.SensorEvent;
import android.hardware.Sensor;
import java.util.List;
public class MainActivity extends Activity {
   SensorManager sm = null;
   TextView textView1 = null;
   List list;
    SensorEventListener sel = new SensorEventListener(){
      public void onAccuracyChanged(Sensor sensor, int accuracy) {}
      public void onSensorChanged(SensorEvent event) {
         float[] values = event.values;
         textView1.setText("x: "+values[0]+"\ny: "+values[1]+"\nz: "+values[2
]);
      }
   };

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

      /* Get a SensorManager instance */
      sm = (SensorManager)getSystemService(SENSOR_SERVICE);

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

      list = sm.getSensorList(Sensor.TYPE_ACCELEROMETER);
      if(list.size()>0){
         sm.registerListener(sel, (Sensor) list.get(0), SensorManager.SENSOR_
DELAY_NORMAL);
      }else{
         Toast.makeText(getBaseContext(), "Error: No Accelerometer.", Toast.
LENGTH_LONG).show();
      }
   }
    @Override
   protected void onStop() {
      if(list.size()>0){
       sm.unregisterListener(sel);
      }
```

```
        super.onStop();
    }
}
```

Output:

Result:

An android application to demonstrate the working of positional sensor successfully

Sensor example program 2:

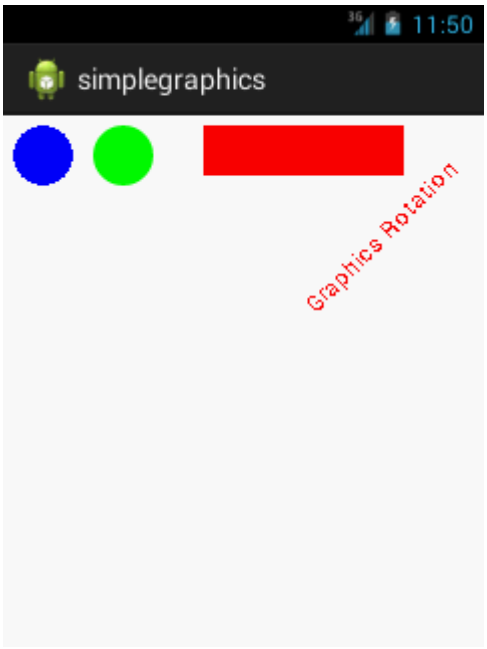
Procedure:

1. Open Eclipse in adk bundle
2. Place a text view control on the interface and give the R.string name.
3. Write the following code in the MainActivity. Java file

```
package com.example.simplegraphics;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;

public class MainActivity extends Activity {

    DemoView demoview;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        demoview = new DemoView(this);
        setContentView(demoview);
    }
```

```
    private class DemoView extends View{
        public DemoView(Context context){
            super(context);
        }

        @Override protected void onDraw(Canvas canvas) {
            super.onDraw(canvas);

            // custom drawing code here
            Paint paint = new Paint();
            paint.setStyle(Paint.Style.FILL);

            // make the entire canvas white
            paint.setColor(Color.WHITE);
            canvas.drawPaint(paint);

            // draw blue circle with anti aliasing turned off
            paint.setAntiAlias(false);
            paint.setColor(Color.BLUE);
            canvas.drawCircle(20, 20, 15, paint);

            // draw green circle with anti aliasing turned on
            paint.setAntiAlias(true);
            paint.setColor(Color.GREEN);
            canvas.drawCircle(60, 20, 15, paint);

            // draw red rectangle with anti aliasing turned off
            paint.setAntiAlias(false);
            paint.setColor(Color.RED);
            canvas.drawRect(100, 5, 200, 30, paint);

            // draw the rotated text
            canvas.rotate(-45);

            paint.setStyle(Paint.Style.FILL);
            canvas.drawText("Graphics Rotation", 40, 180, paint);

            //undo the rotate
            canvas.restore();
        }
    }
    @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;
    }
}
```

Output:



Result:

An android application has been created to demonstrate the working system of positional sensor and executed successfully.