Lab 5
Working with UI Controls - 1

1. Text View

In android, **TextView** is a user interface control that is used to set and display the text to the user based on our requirements. The TextView control will act as like label control and it won't allow users to edit the text.

In android, we can create a TextView control in two ways either in XML layout file or create it in Activity file programmatically.

Android TextView Example

Following is the example of using TextView control in the android application.

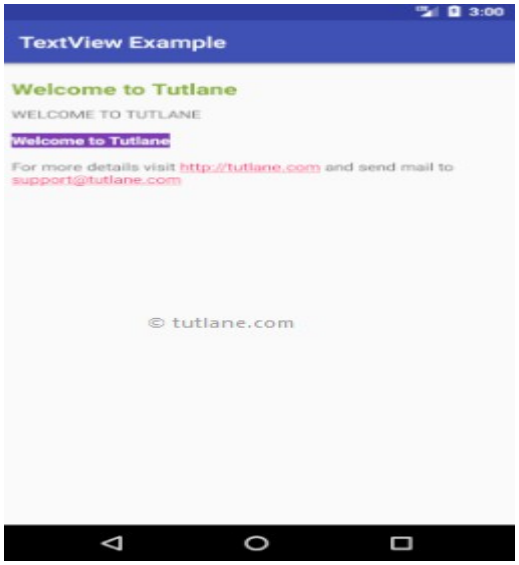
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_marginTop="10dp"
   android:orientation="vertical"
   android:padding="10dp">
   <TextView
      android:id="@+id/textView1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginBottom="10dp"
      android:text="Welcome to Tutlane"
      android:textColor="#86AD33"
      android:textSize="20dp"
      android:textStyle="bold" />
   <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginBottom="15dp"
      android:textAllCaps="true" />
   <TextView
      android:id="@+id/textView3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Welcome to Tutlane"
```

```
        android:textStyle="bold"
        android:textColor="#fff"
        android:background="#7F3AB5"
        android:layout_marginBottom="15dp"/>
    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:autoLink="email|web"
        android:text="For more details visit http://tutlane.com and send mail to support@tutlane.com" />
</LinearLayout>
```

MainActivity.java

```
package com.tutlane.textviewexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView tv = (TextView)findViewById(R.id.textView2);
        tv.setText("Welcome to Tutlane");
    }
}
```



2. Edit Text

In android, EditText is a user interface control which is used to allow the user to enter or modify the text. While using EditText control in our android applications, we need to specify the type of data the text field can accept using the inputType attribute.

For example, if it accept plain text, then we need to specify the inputType as "**text**". In case if **EditText** field is for password, then we need to specify the inputType as "**textPassword**".

Android EditText Control Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="40dp"
   android:orientation="vertical" android:id="@+id/linearlayout" >
   <EditText
      android:id="@+id/txtName"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="25dp"
      android:ems="10"
      android:hint="Name"
      android:inputType="text"
      android:selectAllOnFocus="true" />
   <EditText
      android:id="@+id/txtPwd"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:ems="10"
      android:hint="Password 0 to 9"
      android:inputType="numberPassword" />
   <EditText
      android:id="@+id/txtEmai"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:ems="10"
      android:hint="Email"
      android:inputType="textEmailAddress" />
   <EditText
```

```
        android:id="@+id/txtDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText3"
        android:ems="10"
        android:hint="Date"
        android:inputType="date" />
    <EditText
        android:id="@+id/txtPhone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Phone Number"
        android:inputType="phone"
        android:textColorHint="#FE8DAB"/>
    <Button
        android:id="@+id/btnSend"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="submit"
        android:textSize="16sp"
        android:textStyle="normal|bold" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultView"
        android:layout_marginTop="25dp"
        android:textSize="15dp"/>
</LinearLayout>
```

MainActivity.java

```
package com.tutlane.edittextexample;
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.TextView;
import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity {
    Button btnSubmit;
    EditText name, password, email, dob, phoneno;
    TextView result;
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        name=(EditText)findViewById(R.id.txtName);
        password = (EditText)findViewById(R.id.txtPwd);
        email = (EditText)findViewById(R.id.txtEmai);
        dob = (EditText)findViewById(R.id.txtDate);
        phoneno= (EditText)findViewById(R.id.txtPhone);
        btnSubmit = (Button)findViewById(R.id.btnSend);
        result = (TextView)findViewById(R.id.resultView);
        btnSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (name.getText().toString().isEmpty() || password.getText().toString().isEmpty() ||
email.getText().toString().isEmpty() || dob.getText().toString().isEmpty()
                        || phoneno.getText().toString().isEmpty()) {
                    result.setText("Please Fill All the Details");
                } else {
                    result.setText("Name -  " + name.getText().toString() + " \n" + "Password -  " +
password.getText().toString()
                            + " \n" + "E-Mail -  " + email.getText().toString() + " \n" + "DOB -  " +
dob.getText().toString()
                            + " \n" + "Contact -  " + phoneno.getText().toString());
                }
            }
        });
    }
}
```

3. Button

In android, Button is a user interface control that is used to perform an action whenever the user clicks or tap on it.

Generally, Buttons in android will contain a text or an icon or both and perform an action when the user touches it.

Android Button Control Example

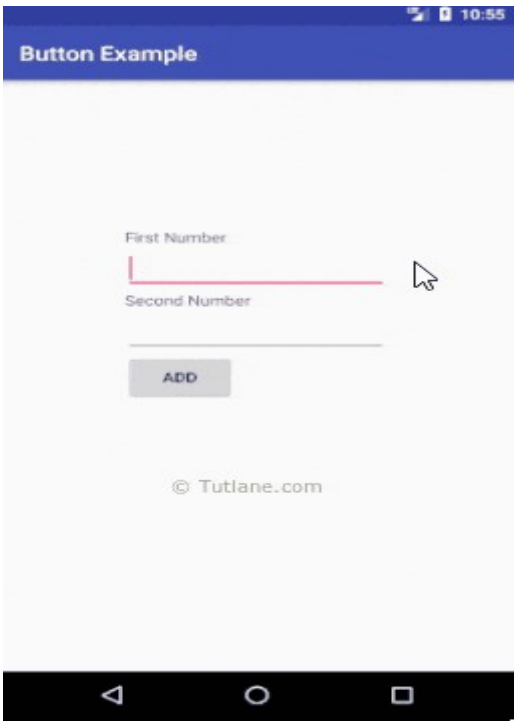
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical" android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:id="@+id/fstTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:layout_marginTop="150dp"
      android:text="First Number" />
   <EditText
      android:id="@+id/firstNum"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:ems="10" />
   <TextView
      android:id="@+id/secTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Second Number"
      android:layout_marginLeft="100dp" />
   <EditText
      android:id="@+id/secondNum"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:ems="10" />
   <Button
      android:id="@+id/addBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```

```
        android:layout_marginLeft="100dp"
        android:text="Add" />
</LinearLayout>
```

MainActivity.java

```
package com.tutlane.buttonexample;
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 {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final EditText firstNum = (EditText)findViewById(R.id.firstNum);
        final EditText secNum = (EditText)findViewById(R.id.secondNum);
        Button btnAdd = (Button)findViewById(R.id.addBtn);
        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(firstNum.getText().toString().isEmpty() || secNum.getText().toString().isEmpty())
                {
                    Toast.makeText(getApplicationContext(), "Please fill all the fields",
Toast.LENGTH_SHORT).show();
                }
                else {
                    int num1 = Integer.parseInt(firstNum.getText().toString());
                    int num2 = Integer.parseInt(secNum.getText().toString());
                    Toast.makeText(getApplicationContext(), "SUM = " + (num1 + num2),
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```



4. Image Button

In android, Image Button is a user interface control that is used to display a button with an image and to perform an action when a user clicks or taps on it.

By default, the ImageButton looks same as normal button and it performs an action when a user clicks or touches it, but the only difference is we will add a custom image to the button instead of text.

Following is the pictorial representation of using **Image Buttons** in android applications.

Android ImageButton Control Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical" android:layout_width="match_parent"
   android:layout_height="match_parent" android:id="@+id/l_layout">
   <TextView
      android:id="@+id/fstTxt"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="150dp"
        android:text="First Number"/>
    <EditText
        android:id="@+id/firstNum"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10"/>
    <TextView
        android:id="@+id/secTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Number"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/secondNum"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10" />
    <ImageButton
        android:id="@+id/addBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:src="@drawable/add_icon" />
</LinearLayout>
```

MainActivity.java

```
package com.tutlane.buttonexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

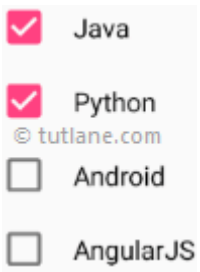
```
        setContentView(R.layout.activity_main);
        final EditText firstNum = (EditText)findViewById(R.id.firstNum);
        final EditText secNum = (EditText)findViewById(R.id.secondNum);
        ImageButton btnAdd = (ImageButton)findViewById(R.id.addBtn);
        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(firstNum.getText().toString().isEmpty() || secNum.getText().toString().isEmpty())
                {
                    Toast.makeText(getApplicationContext(), "Please fill all the fields",
Toast.LENGTH_SHORT).show();
                }
                else {
                    int num1 = Integer.parseInt(firstNum.getText().toString());
                    int num2 = Integer.parseInt(secNum.getText().toString());
                    Toast.makeText(getApplicationContext(), "SUM = " + (num1 + num2),
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

5. Check Box

In android, **CheckBox** is a two-states button that can be either checked (ON) or unchecked (OFF) and it will allow users to toggle between the two states (ON / OFF) based on the requirements.

Generally, we can use multiple **CheckBox** controls in android application to allow users to select one or more options from the set of values.

Following is the pictorial representation of using **CheckBox** control in android applications.



By default, the android **CheckBox** will be in the **OFF** (**Unchecked**) state. We can change the default state of CheckBox by using **android:checked** attribute.

In case, if we want to change the state of **CheckBox** to **ON** (**Checked**), then we need to set **android:checked = "true"** in our XML layout file.

Android CheckBox Control Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical" android:layout_width="match_parent"
   android:layout_height="match_parent">
   <CheckBox
      android:id="@+id/chkJava"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="10dp"
      android:layout_marginTop="150dp"
      android:layout_marginLeft="100dp"
      android:text="Java"
      android:onClick="onCheckboxClicked"/>
   <CheckBox
      android:id="@+id/chkPython"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_marginLeft="100dp"
        android:text="Python"
        android:onClick="onCheckboxClicked"/>
    <CheckBox
        android:id="@+id/chkAndroid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_marginLeft="100dp"
        android:text="Android"
        android:onClick="onCheckboxClicked"/>
    <CheckBox
        android:id="@+id/chkAngular"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_marginLeft="100dp"
        android:text="AngularJS"
        android:onClick="onCheckboxClicked"/>
    <Button
        android:id="@+id/getBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:text="Get Details" />
</LinearLayout>
```

MainActivity.java

```
package com.tutlane.checkboxexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    CheckBox android, java, angular, python;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
        android = (CheckBox)findViewById(R.id.chkAndroid);
        angular = (CheckBox)findViewById(R.id.chkAngular);
        java = (CheckBox)findViewById(R.id.chkJava);
        python = (CheckBox)findViewById(R.id.chkPython);
        Button btn = (Button)findViewById(R.id.getBtn);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String result = "Selected Courses";
                if(android.isChecked()){
                result += "\nAndroid";
                }
                if(angular.isChecked()){
                    result += "\nAngularJS";
                }
                if(java.isChecked()){
                    result += "\nJava";
                }
                if(python.isChecked()){
                    result += "\nPython";
                }
                Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
            }
        });
    }
    public void onCheckboxClicked(View view) {
        boolean checked = ((CheckBox) view).isChecked();
        String str="";
        // Check which checkbox was clicked
        switch(view.getId()) {
            case R.id.chkAndroid:
                str = checked?"Android Selected":"Android Deselected";
                break;
            case R.id.chkAngular:
                str = checked?"AngularJS Selected":"AngularJS Deselected";
                break;
            case R.id.chkJava:
                str = checked?"Java Selected":"Java Deselected";
                break;
            case R.id.chkPython:
                str = checked?"Python Selected":"Python Deselected";
                break;
        }
        Toast.makeText(getApplicationContext(), str, Toast.LENGTH_SHORT).show();
    }
}
```

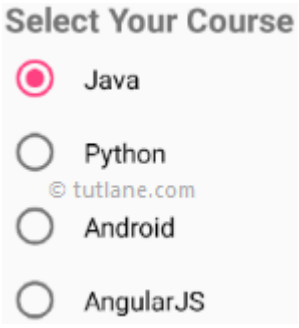
5. Radio Button

In android, **Radio Button** is a two-states button that can be either checked or unchecked and it's the same as CheckBox control, except that it will allow only one option to select from the group of options.

The user can press or click on the radio button to make it select. In android, CheckBox control allow users to change the state of control either Checked or Unchecked but the radio button cannot be unchecked once it is checked.

Generally, we can use **RadioButton** controls in an android application to allow users to select only one option from the set of values.

Following is the pictorial representation of using **RadioButton** control in android applications.



In android, we use radio buttons with in a **RadioGroup** to combine multiple radio buttons into one group and it will make sure that users can select only one option from the group of multiple options.

By default, the android **RadioButton** will be in **OFF** (**Unchecked**) state. We can change the default state of **RadioButton** by using **android:checked** attribute.

In case, if we want to change the state of **RadioButton** to **ON** (**Checked**), then we need to set **android:checked = "true"** in our XML layout file.

Android RadioButton Control Example

activity_main.xml

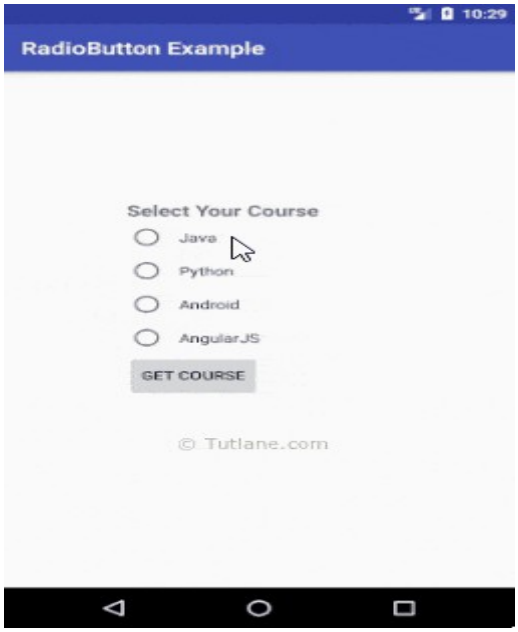
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="150dp"
      android:layout_marginLeft="100dp"
      android:textSize="18dp"
      android:text="Select Your Course"
      android:textStyle="bold"
      android:id="@+id/txtView"/>
<RadioGroup
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="vertical"
   android:id="@+id/rdGroup"
   android:layout_below="@+id/txtView">
   <RadioButton
      android:id="@+id/rdbJava"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="10dp"
      android:layout_marginLeft="100dp"
      android:text="Java"
      android:onClick="onRadioButtonClicked"/>
   <RadioButton
      android:id="@+id/rdbPython"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```

```
        android:padding="10dp"
        android:layout_marginLeft="100dp"
        android:text="Python"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton
        android:id="@+id/rdbAndroid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_marginLeft="100dp"
        android:text="Android"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton
        android:id="@+id/rdbAngular"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_marginLeft="100dp"
        android:text="AngularJS"
        android:onClick="onRadioButtonClicked"/>
</RadioGroup>
    <Button
        android:id="@+id/getBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_below="@+id/rdGroup"
        android:text="Get Course" />
</RelativeLayout>
```

MainActivity.java

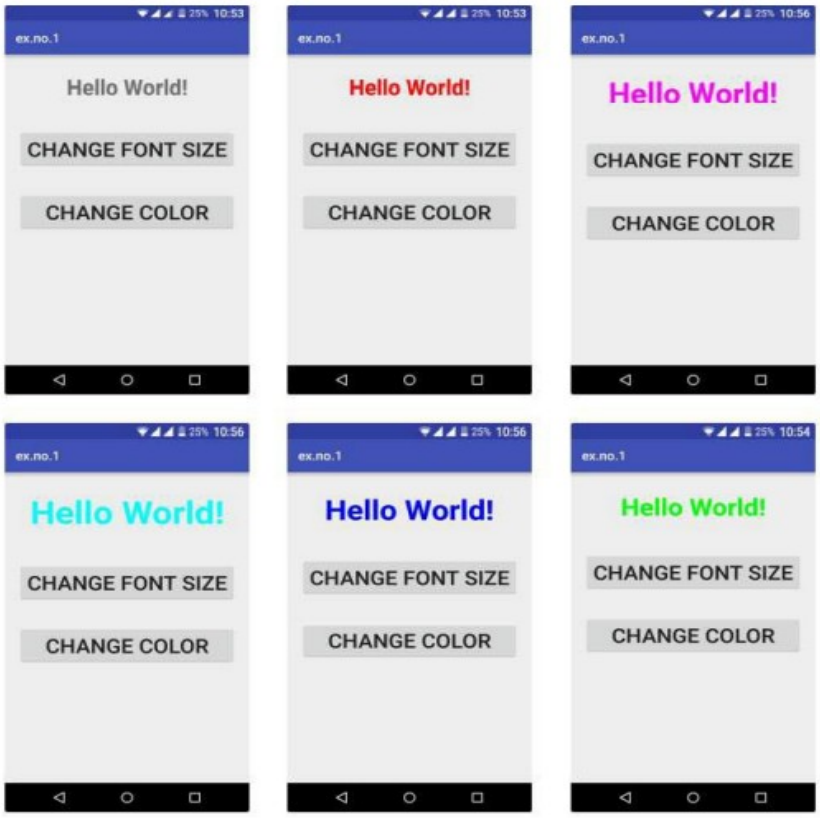
```
package com.tutlane.radiobuttonexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    RadioButton android, java, angular, python;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
        android = (RadioButton)findViewById(R.id.rdbAndroid);
        angular = (RadioButton)findViewById(R.id.rdbAngular);
        java = (RadioButton)findViewById(R.id.rdbJava);
        python = (RadioButton)findViewById(R.id.rdbPython);
        Button btn = (Button)findViewById(R.id.getBtn);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String result = "Selected Course: ";
                result+= (android.isChecked())?"Android":(angular.isChecked())?"AngularJS":
(java.isChecked())?"Java":(python.isChecked())?"Python":"";
                Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
            }
        });
    }
    public void onRadioButtonClicked(View view) {
        boolean checked = ((RadioButton) view).isChecked();
        String str="";
        // Check which radio button was clicked
        switch(view.getId()) {
            case R.id.rdbAndroid:
                if(checked)
                str = "Android Selected";
                break;
            case R.id.rdbAngular:
                if(checked)
                str = "AngularJS Selected";
                break;
            case R.id.rdbJava:
                if(checked)
                str = "Java Selected";
                break;
            case R.id.rdbPython:
                if(checked)
                str = "Python Selected";
                break;
        }
        Toast.makeText(getApplicationContext(), str, Toast.LENGTH_SHORT).show();
    }
}
```

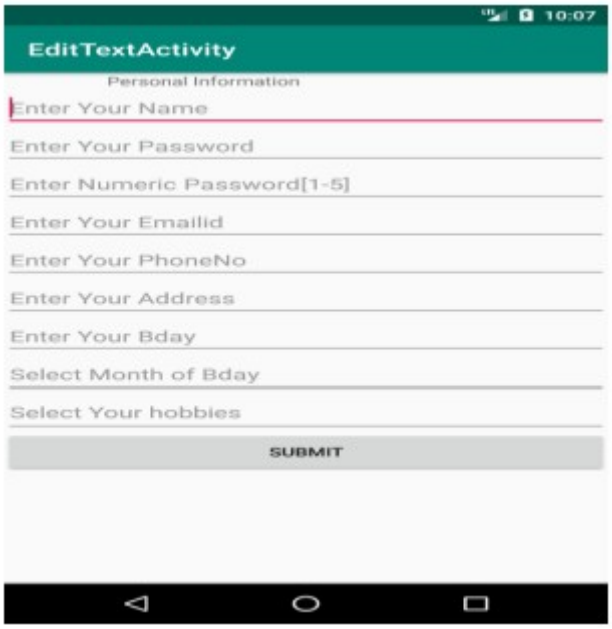


Exercise

1. Create an application that uses GUI components, Font and Color. In this application you have one textview control to display Hello World!, two buttons. One button is change font size and second button is change color.



2. Create an application which show various types of EditText control and Toast the value of each EditText on click of the Button.



3. Write an application to create simple calculator as shown in figure below.

4. Create an Activity having three Radio Button group. One for color, one for alignment and one for Text size perform necessary changes according to Radio button selected.