

### Practical No. 9: Develop a program to implement Button, Image Button and Toggle Button.

#### I. Practical Significance

In this practical, UI controls in android like Buttons are studied. There are various types of buttons like Image button and toggle button which is studied.

#### II. Relevant Program Outcomes (POs)

- PO 1. Basic knowledge
- PO 2. Discipline knowledge
- PO 3. Experiments and practice
- PO 4. Engineering tools
- PO 7. Ethics
- PO 10. Life-long learning

#### III. Competency and Practical Skills

*“Create simple Android applications.”*

This practical is expected to develop the following skills

- 1. Able to develop UI controls like various types of buttons.
- 2. Able to use buttons which handles events.
- 3. Able to build Passive and Active UI controls.

#### IV. Relevant Course Outcome(s)

- 1. Develop rich user Interfaces by using layouts and controls.
- 2. Use User Interface components for android application development.

#### V. Practical Outcome (PrOs)

Develop a program to implement Button, Image Button and Toggle Button.

#### VI. Relevant Affective Domain Related Outcome(s)

- 1. Work collaboratively in team
- 2. Follow ethical Practices.

#### VII. Minimum Theoretical Background

##### 1. Buttons-

In Android, Button represents a push button. The android.widget.Button is subclass of TextView class and Compound Button is the subclass of Button class. A Push buttons can be clicked, or pressed by the user to perform an action. There are different types of buttons used in android such as Compound Button, Toggle Button, Radio Button. Button is a subclass of TextView class and compound button is the subclass of Button class. On a button we can perform different actions or events like click event, pressed event, touch event etc. Android buttons are GUI components which are sensible to taps (clicks) by the user. When the user taps/clicks on button in an Android app, the app can respond to the click/tap. These buttons can be divided into two categories: the first is Buttons with text on, and second is buttons with an image on.

**VIII. Resources required (Additional)**

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

**IX. Practical related Questions**

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write a piece of code to set id of the button.
2. How to add image to resources file?
3. List four Android Toggle Button control attributes.

(Space for answers)

IX

1) &lt;Button

```
    android:id="@+id/btn1"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content">
</Button>
```

2) To add the image to resource file -

i) place your image to a resource file in  
res/drawable directory of your android project.

ii) If you have image specific to screen densities you can use respective directories - hdpi.drawable .

iii) Access these image in your code using  
R.drawable.image\_name or in xml using  
@drawable/.image\_name'

3) i) android:textOn      iii) android:checked  
ii) android:textOff      iv) android:background

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

- (Use blank space for answers or attach more pages if needed)
1. Write a program to create a toggle button to display ON / OFF Bluetooth on the display screen.
  2. Write a program to create a simple calculator.

(Space for answers)

1) ActivityMain.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/
    android"
    android:id="@+id/l1"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/t1"
        android:height="wrap_content"
        android:layout_width="wrap_content"
        android:textSize="12dp">
    </TextView>
    <ToggleButton
        android:id="@+id/tb1"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:onClick="onToggleClick">
    </ToggleButton>
</LinearLayout>
```

```
MainActivity.java
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.ToggleButton;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity
{
    ToggleButton t1;
    TextView tw;
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        t1 = (ToggleButton) findViewById(R.id.b1);
        tw = (TextView) findViewById(R.id.t1);
    }

    public void onToggleClick(View view)
    {
        if(t1.isChecked())
            tw.setText("Toggle Button is ON");
        else
            tw.setText("Toggle Button is OFF");
    }
}
```

Simple calculator -  
ActivityMain.xml

```
<? xml version="1.0" encoding="UTF-8" ?>
< LinearLayout
    xmlns: android = " http://schemas.android.com/apk/
        android.
    android: id = "@+id/ll"
    android: layout_height = "match_parent"
    android: layout_width = "match_parent"
    android: orientation = "vertical" >
    < EditText
        android: id = "@+id/e1"
        android: layout_height = "wrap_content"
        android: layout_width = "match_parent"
        android: layout_gravity = "center_horizontal"
        android: textSize = "50dp"
        android: hint = "Enter Value here" >
    </EditText>
    < EditText
        android: id = "@+id/e2"
        android: layout_height = "wrap_content"
        android: layout_width = "match_parent"
        android: layout_gravity = "center_horizontal"
        android: textSize = "50dp"
        android: hint = "Enter Value here" >
    </EditText>
    < Button
        android: id = "@+id/b1"
        android: layout_height = "wrap_content"
        android: layout_width = "match_parent"
        android: text = "+"
        android: textSize = "100dp" >
    </Button>
```

```
<Button  
    android: id = "@+id/b2"  
    android: layout_height = "wrap_content"  
    android: layout_width = "match_parent"  
    android: text = "-"  
    android: textSize = "150 dp" >  
</Button>  
<Button  
    android: id = "@+id/b3"  
    android: layout_height = "wrap_content"  
    android: layout_width = "match_parent"  
    android: text = "*"  
    android: textSize = "100 dp" >  
</Button>  
<Button  
    android: id = "@+id/b4"  
    android: layout_height = "wrap_content"  
    android: layout_width = "match_parent"  
    android: text = "/"  
    android: textSize = "100 dp" >  
</Button>  
</LinearLayout>  
<TextView  
    android: id = "@+id/t1"  
    android: layout_height = "wrap_content"  
    android: layout_width = "wrap_content"  
    android: textSize = "100 dp" >  
</TextView>  
</LinearLayout>
```

### MainActivity.java

```
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
android.appcompat.app.AppCompatActivity;  
public class MainActivity extends AppCompatActivity
```

```
    EditText e1, e2;
```

```
    TextView t1;
```

```
    Button b1, b2, b3, b4;
```

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

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

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

```
    e1 = (EditText) findViewById(R.id.e1);
```

```
    e2 = (EditText) findViewById(R.id.e2);
```

```
    t1 = (TextView) findViewById(R.id.t1);
```

```
    b1 = (Button) findViewById(R.id.b1);
```

```
    b2 = (Button) findViewById(R.id.b2);
```

```
    b3 = (Button) findViewById(R.id.b3);
```

```
    b4 = (Button) findViewById(R.id.b4);
```

```
b1.setOnClickListener(new View.OnClickListener())
```

```
{
```

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

```
        float n1 = Float.parseFloat(e1.getText());
```

```
        float n2 = Float.parseFloat(e2.getText());
```

Date \_\_\_\_\_

```
float result;
result = n1 + n2;
T1.setText("Result is " + result);
}

b2.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        float n1 = Float.parseFloat(e1.getText());
        float n2 = Float.parseFloat(e2.getText());
        float result;
        result = n1 - n2;
        T1.setText("Subtraction is " + result);
    }
});

b3.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        float n1 = Float.parseFloat(e1.getText());
        float n2 = Float.parseFloat(e2.getText());
        float result;
        result = n1 * n2;
        T1.setText("Multiplication is " + result);
    }
});

b4.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        float n1 = Float.parseFloat(e1.getText());
        float n2 = Float.parseFloat(e2.getText());
        float result = n1 / n2;
        T1.setText("Division is " + result);
    }
});
```

**Practical No. 10: Develop a program to implement login window using above UI controls.**

**I. Practical Significance**

In this practical, all the previous UI controls in android like Text View, Edit Text Buttons which are studied are implemented in this practical. Events are also handled on the android UI controls used in the practical.

**II. Relevant Program Outcomes (POs)**

- PO 1. Basic knowledge
- PO 2. Discipline knowledge PO 4. Engineering tools
- PO 7. Ethics
- PO 10. Life-long learning

**III. Competency and Practical Skills**

“Create simple Android applications.”

This practical is expected to develop the following skills

- 1. Able to use the layout managers.
- 2. Able to develop android UI controls to create login window without using databases.

**IV. Relevant Course Outcome(s)**

- 1. Develop rich user Interfaces by using layouts and controls.
- 2. Use User Interface components for android application development.

**V. Practical Outcome (PrOs)**

Develop a program to implement Button, Image Button and Toggle Button.

**VI. Relevant Affective Domain Related Outcome(s)**

- 1. Work collaboratively in team
- 2. Follow ethical Practices.

**VII. Minimum Theoretical Background**

A login application is the screen asking your credentials to login to some particular application. You might have seen it when logging into facebook, twitter etc. Define two Text View asking username and password of the user. The password Text View must have input Type set to password. Its syntax is given below

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Name the file in which respective XML components can be added.
2. List all the UI components which can be used to develop login window.

(Space for answers)

1 ① activity\_main.xml  
② string.xml  
③ dimension.xml

② ① TextView  
② EditText  
③ Button

## X. Exercise

*Note: Faculty must ensure that every group of students use different input value.*

(Use blank space for answers or attach more pages if needed)

1. Write a program to create a login form for a social networking site.
2. Write a program to create a login form for student registration system

(Space for answers)

```

    1. activity_main.xml
        <?xml version = "1.0" encoding = "utf-8"?>
        <LinearLayout xmlns:android = "http://schemas.
        android.com/apk/res/android"
        android:layout_height = "match_parent"
        android:layout_width = "match_parent"
        android:orientation = "vertical"

        <TextView>
            android:id = "@+id/t1"
            android:layout_height = "wrap_content"
            android:layout_width = "wrap_content"
            android:text = "Login Form"
            android:textSize = "40dp"
            android:layout_gravity = "center"
        </>

        <EditText>
            android:id = "@+id/E1"
            android:layout_height = "wrap_content"
            android:layout_width = "match_parent"
            android:hint = "Username"
            android:inputType = "text"
        </>

        <EditText>
            android:id = "@+id/E2"
            android:layout_height = "wrap_content"
            android:layout_width = "match_parent"
        
```

```
    android: hint = "password"
    android: inputType = "textPassword"
  />
<Button
    android: id = "@+id/b1"
    android: layout_height = "wrap_content"
    android: layout_width = "wrap_content"
    android: text = "Login"
    android: textSize = "40dp"
    android: layout_gravity = "center"
/>
```

• 1. MainActivity.java

```
package com.example.networkexample;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.*;
```

```
public class MainActivity extends AppCompatActivity
```

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

```
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    EditText e1 = findViewById(R.id.E1);
    EditText e2 = findViewById(R.id.E2);
    Button b1 = findViewById(R.id.b1);
```

```
    b1.setOnClickListener(new View.OnClickListener()
```

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

Date \_\_\_\_\_

```
String username = e1.getText();
String password = e2.getText();

if (username.equals("GPAN") && password.equals
    ("networking"))
{
    Toast.makeText(MainActivity.this, "Login
        Successful", Toast.LENGTH_LONG).show();
}
else
{
    Toast.makeText(MainActivity.this, "Login Fail".
        Toast.LENGTH_LONG).show();
}
});
```

2) Login Form for student registration system

activity\_main.xml

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

    <TextView
        android:id="@+id/t1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Student Registration System"
        android:textSize="40dp"
        android:layout_gravity="center"
    />

    <EditText
        android:id="@+id/et1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:inputType="text"
    />

    <EditText
        android:id="@+id/e2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enrollment No."
        android:inputType="number"
    />
```

```
<EditText  
    android: id = "@+id/le3"  
    android: layout_width = "match_parent"  
    android: layout_height = "wrap_content"  
    android: hint = "Class"  
    android: inputType = "text"  
/>  
  
<EditText  
    android: id = "@+id/le4"  
    android: layout_width = "match_parent"  
    android: layout_height = "wrap_content"  
    android: hint = "Department"  
    android: inputType = "text"  
/>  
  
<Button  
    android: id = "@+id/b1"  
    android: layout_width = "wrap_content"  
    android: layout_height = "wrap_content"  
    android: text = "Login"  
    android: textSize = "40dp"  
    android: layout_gravity = "center"  
/>  
-</LinearLayout>  
<TextView  
    android: id = "@+id/t2"  
    android: layout_width = "wrap_content"  
    android: layout_height = "wrap_content"  
    android: textSize = "40dp"  
    android: layout_gravity = "center"  
/>  
-</LinearLayout>
```

### MainActivity.java

```
package com.example.registrationexample;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

public class MainActivity extends AppCompatActivity
{
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        EditText username = findViewById(R.id.e1);
        EditText enrollment = findViewById(R.id.e2);
        EditText classs = findViewById(R.id.e3);
        EditText dept = findViewById(R.id.e4);
        Button b1 = findViewById(R.id.b1);
        TextView T1 = findViewById(R.id.T2);

        b1.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
                if(string.username1 == username.getText().isEmpty())
                    || string.password1 == password.getText().isEmpty()
                    || string.class1 == classs.getText().isEmpty()
                    || string.dept1 == dept.getText().isEmpty()
                )
                {
                    T1.setText("please Enter All the Details");
                }
                else
                {
                }
            }
        });
    }
}
```

Date \_\_\_\_\_

```
T1.setText("Registration Successful");
```

5

3. How do you feel about your MySpace or other social networking site?

); *legume*) in A. *agg.* *leguminosae*. *Chlorophytum* *desmodii*

1

*Leptobium bicolor* fragariae

1

*ptivitologan)ggA thorax ptivitologan umi sudiug*

Colorless or yellowish-green, often with a bluish tinge; thick, brittle, fibrous.

Digitized by Google

(gimp viivibp) suunt 2.5m/100(7.2)19

(1a, b) *b7c* *b7d* *b7e* *b7f* *b7g* *b7h* *b7i* *b7j* *b7k* *b7l* *b7m* *b7n* *b7o* *b7p* *b7q* *b7r* *b7s* *b7t* *b7u* *b7v* *b7w* *b7x* *b7y* *b7z*

(13) *hi a) bTugqas/ ba'q + Inaqillorsa* *ixqTaqib3*

(89-81-8) *brevioris* *baiti*: *wolf* " *taoT4013*

~~(eg. big blue ball = bush bush)~~

(Ad. bi. 8) brasil(hois-sa) contin

• *Georgi Gurdjieff's "Law of Seven" is a system of spiritual development based on the concept of seven levels of consciousness.*

*With many thanks to Dr. G. R. Thompson for his kind permission to publish this note.*

10. The following table shows the number of hours worked by 1000 workers in a certain industry.

• *(X) 2011/2012 bior. 2011/2012*

### Practical No. 11: Develop a program to implement Checkbox.

#### I. Practical Significance

Android **CheckBox** is a type of two state button either checked or unchecked. There can be a lot of usage of checkboxes. For example, it can be used to know the hobby of the user, activate/deactivate the specific action etc.

#### II. Relevant Program Outcomes (POs)

- PO 2. Discipline knowledge
- PO 3. Experiments and practice
- PO 4. Engineering tools
- PO 10. Life-long learning

#### III. Competency and Practical Skills

"Create simple Android applications."

This practical is expected to develop the following skills

- 1. Able to develop an application using Checkbox control.

#### IV. Relevant Course Outcome(s)

Use User Interface components for android application development.

#### V. Practical Outcomes (PrOs)

Develop a program to implement Checkbox.

#### VI. Relevant Affective Domain related Outcome(s)

- 1. Work collaboratively in team
- 2. Follow ethical practices

#### VII. Minimum Theoretical Background

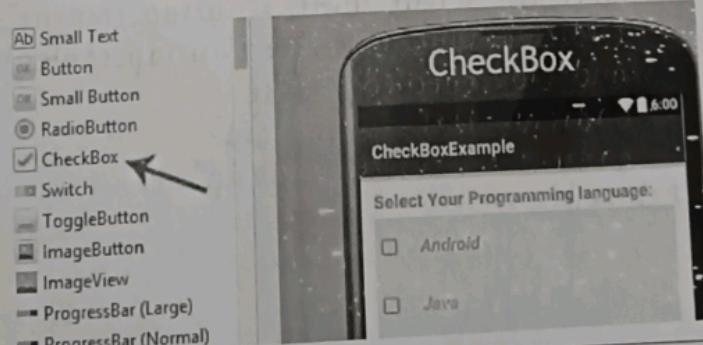
Android Checkbox class is the subclass of Compound Button class.

##### Methods of Checkbox class

There are many inherited methods of View, Text View, and Button classes in the Checkbox class. Some of them are as follows:

- public boolean isChecked() : Returns true if it is checked otherwise false.
- public void setChecked(boolean status) : Changes the state of the Checkbox.

Following figure shows different checkboxes



54

**VIII. Resources required (Additional)**

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

**IX. Practical related Questions**

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Name the different methods of Checkbox.
2. List different attributes of Checkbox.
3. Write xml tag to create a checkbox named "Android".

(Space for answers)

1] Methods of checkbox -

- i) public boolean isChecked()
- ii) public void setChecked(boolean status)

2] Attributes of checkbox -

- i) id
- ii) text
- iii) checked
- iv) gravity
- v) background

3] &lt; checkBox

```

    android: id = "@+id/c1"
    android: layout_width = "wrap_content"
    android: layout_height = "wrap_content"
    android: checked = "true"
    android: text = "Android"
    android: textSize = "40dp" >
</checkBox>
```

X. Exercise

Note: Faculty must ensure that every group of students use different examples.  
(Use blank space for answers or attach more pages if needed)

1. Write a program to show five checkboxes and toast selected checkboxes.

(Space for answers)

```
1] 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:orientation="vertical">
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="MAD"
        android:textSize="50 dp"
        android:id="@+id/c1"/>
    </CheckBox>
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="PHP"
        android:textSize="50 dp"
        android:id="@+id/c2"/>
    </CheckBox>
    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Python"
        android:textSize="50 dp"
        android:id="@+id/c3"/>
    </CheckBox>

```

```
<CheckBox  
    android:layout_width = "wrap_content"  
    android:layout_height = "wrap_content"  
    android:text = "ETI"  
    android:textSize = "50.dp"  
    android:id = "@+id/c5"  
/>  
<CheckBox  
    android:layout_width = "wrap_content"  
    android:layout_height = "wrap_content"  
    android:text = "EDE"  
    android:textSize = "50.dp"  
    android:id = "@+id/c5"  
/>  
</LinearLayout>
```

Date \_\_\_\_\_

### MainActivity.xml

```
package com.example.checkboxexample;  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.*;
```

```
public class MainActivity extends AppCompatActivity  
    implements View.OnClickListener
```

{

```
    CheckBox c1, c2, c3, c4, c5;
```

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

{

```
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        c1 = (CheckBox) findViewById(R.id.c1);  
        c2 = (CheckBox) findViewById(R.id.c2);  
        c3 = (CheckBox) findViewById(R.id.c3);  
        c4 = (CheckBox) findViewById(R.id.c4);  
        c5 = (CheckBox) findViewById(R.id.c5);
```

```
        c1.setOnClickListener(this);
```

```
        c2.setOnClickListener(this);
```

```
        c3.setOnClickListener(this);
```

```
        c4.setOnClickListener(this);
```

```
        c5.setOnClickListener(this);
```

}

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

{

```
    StringBuffer str = new StringBuffer(" you have  
    clicked: ");
```

```
    if (c1.isChecked())
```

{

str.append(" MAD");  
if (c2.isChecked())  
{  
 str.append(" PHP");  
}  
if (c3.isChecked())  
{  
 str.append(" python");  
}  
if (c4.isChecked())  
{  
 str.append(" ETI");  
}  
if (c5.isChecked())  
{  
 str.append(" EDE");  
}  
Toast.makeText(MainActivity.this, str, Toast.LENGTH\_LONG).show();

### Practical No. 12: Develop a program to implement Radio Button and Radio Group.

#### I. Practical Significance

Radio Buttons are used when we need to select only one item from a list of presented items. If Radio Buttons are in Radio Group, when one Radio Button within a group is selected, all others are automatically deselected.

#### II. Relevant Program Outcomes (POs)

- PO 2. Discipline knowledge
- PO 3. Experiments and practice
- PO 4. Engineering tools

#### III. Competency and Practical Skills

"Create simple Android applications."

This practical is expected to develop the following skills

- 1. Able to develop an application using Radio Button and Radio Group controls.

#### IV. Relevant Course Outcome(s)

- 1. Develop rich user Interfaces by using layouts and controls.
- 2. Use User Interface components for android application development.

#### V. Practical Outcomes (PrOs)

Use Develop a program to implement Radio Button and Radio Group.

#### VI. Relevant Affective Domain Related Outcome(s)

- 1. Work collaboratively in team
- 2. Follow ethical practices

#### VII. Minimum Theoretical Background

Radio Button is generally used with *Radio Group*. **Radio Group** is a set of radio buttons, marking one radio button as checked makes all other radio buttons as unchecked. A radio button consists of two states – checked and unchecked. Clicking an unchecked button changes its state to "checked" state and "unchecked" for the previously selected radio button. To toggle a checked state to unchecked state, we need to choose another item.

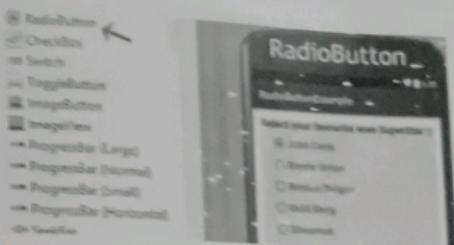
Following are the important attributes related to Radio Group control.

- 1. **android:checkedButton** : This is the id of child radio button that should be checked by default within this radio group.
- 2. **android:orientation** : This property on the Radio group defines the orientation to position its child view consisting of Radio Buttons.

Following are the few methods of radio button:

- 1. **check(id)**: This sets the selection to the radio button whose identifier is passed in parameter. -1 is used as the selection identifier to clear the selection.
- 2. **clearCheck()** : It clears the selection. When the selection is cleared, no radio button in this group is selected and getCheckedRadioButtonId() returns null.
- 3. **getCheckedRadioButtonId()** : It returns the identifier of the selected radio button in this group. If its empty selection, the returned value is-1.

4. `setOnCheckedChangeListener()` : This registers a callback to be invoked when the checked radio button changes in this group. We must supply instance of Radio Group. OnCheckedChangeListener to setOnCheckedChangeListener() method



### VIII. Resources used (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

### IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write xml tag to create a Radio button.
2. Write the purpose of Radio Button
3. List different methods of Radio Button

(Space for answers)

1) <RadioButton  
 android:id="@+id/r1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="First Radio Button"

1>

2) Radio Buttons are used to select only one option from the list of multiple available options.

- 3)
  - (i) check (id)
  - (ii) getCheckedRadioButtonId ()
  - (iii) setOnCheckedChangeListener ()

#### X. Exercise

*Note: Faculty must ensure that every group of students use different examples.*

(Use blank space for answers or attach more pages if needed)

1. Write a program to show the following output. First two radio buttons are without using radio group and next two radio buttons are using radio group. Note the changes between these two. Also toast which radio button has been selected.



Single Radio Buttons

- Radio Button 1  
 Radio Button 2

Radio button inside RadioGroup

- Male  
 Female

**SHOW SELECTED**

(Space for answers)

1) 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:orientation="vertical">
<TextView
    android:id="@+id/t1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Single Radio Button"
    android:textSize="40dp"
    android:layout_gravity="center"/>
</RadioGroup>
<RadioButton
    android:id="@+id/R1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Radio Button 1"
    android:textSize="40dp"/>
</>
<RadioButton
    android:id="@+id/R2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Radio Button 2"
    android:textSize="40dp"/>
</>
</RadioGroup>
<TextView>
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android: text = " Radio Buttons inside RadioGroup"
    android: textSize = " 40 dp"
    android: layout_gravity = " center"
/>
< RadioGroup
    android: layout_width = " wrap_content"
    android: layout_height = " wrap_content"
< RadioButton
    android: id = "@+id/R3"
    android: layout_width = " wrap_content"
    android: layout_height = " wrap_content"
    android: text = " Male"
    android: textSize = " 40 dp"
/>
< RadioButton
    android: id = "@+id/R4"
    android: layout_width = " wrap_content"
    android: layout_height = " wrap_content"
    android: text = " Female"
    android: textSize = " 40 dp"
/>
</ RadioGroup>
< Button
    android: id = "@+id/ B1"
    android: layout_width = " wrap_content"
    android: layout_height = " wrap_content"
    android: text = " Show Selected"
    android: textSize = " 40 dp"
/>
</ LinearLayout>
```

```
MainActivity.java
package com.example.radioexample;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.*;
public class MainActivity extends AppCompatActivity
{
    RadioButton R1, R2, R3, R4;
    String selected;
    Button b1;
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        R1 = (RadioButton) findViewById(R.id.R1);
        R2 = (RadioButton) findViewById(R.id.R2);
        R3 = (RadioButton) findViewById(R.id.R3);
        R4 = (RadioButton) findViewById(R.id.R4);
        b1 = (Button) findViewById(R.id.B1);
        b1.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
                selected = " ";
                if (R1.isChecked())
                {
                    selected = R1.getText().toString() + "\n";
                }
                if (R2.isChecked())
                {
                    selected = R2.getText().toString() + "\n";
                }
            }
        });
    }
}
```

Date

```
selected = R2.getText().toString() + "\n";
if (R3.isChecked())
    selected = R3.getText().toString() + "\n";
if (R4.isChecked())
    selected = R4.getText().toString() + "\n";
Toast.makeText(MainActivity.this, selected,
    Toast.LENGTH_LONG).show();
});
```