Universiti Malaysia PAHANG
Engineering • Technology • Creativity

BCS3283-Mobile Application Development

Chapter 5
How to program buttons

Editor
Dr. Mohammed Falah Mohammed

Faculty of Computer Systems & Software Engineering
falah@ump.edu.my

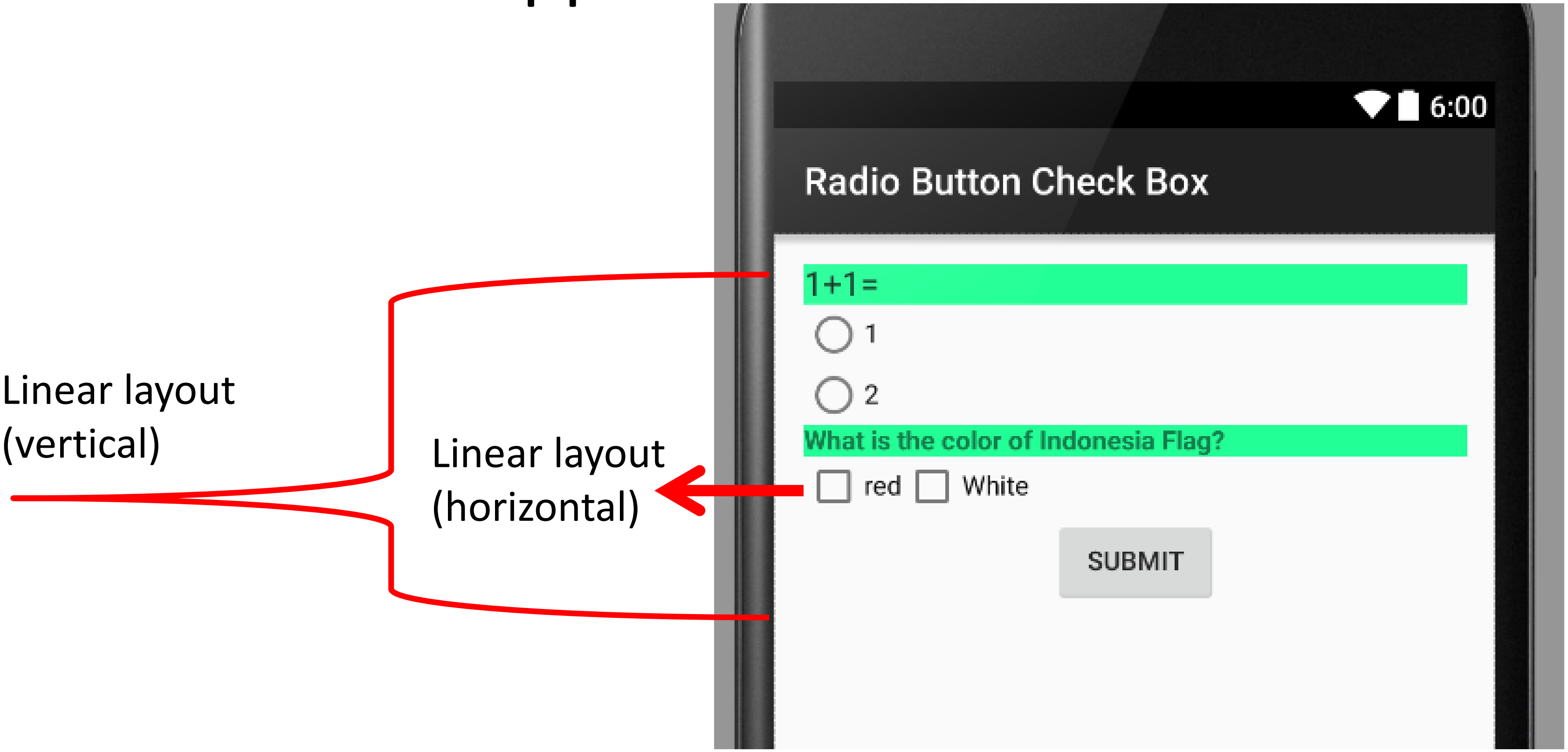
How to program buttons

- Aims
  - To learn how to program a Radio Button and Check Box in android studio.

- Expected Outcomes
  - Ability to create a Radio Button.
  - Ability to create a Check Box Button
  - Ability to manage and deal with Linear Layouts (vertical and horizontal)

Quiz App

- Create new application with GUI:



Linear layout (vertical)

Linear layout (horizontal)

Radio Button Check Box

1+1=

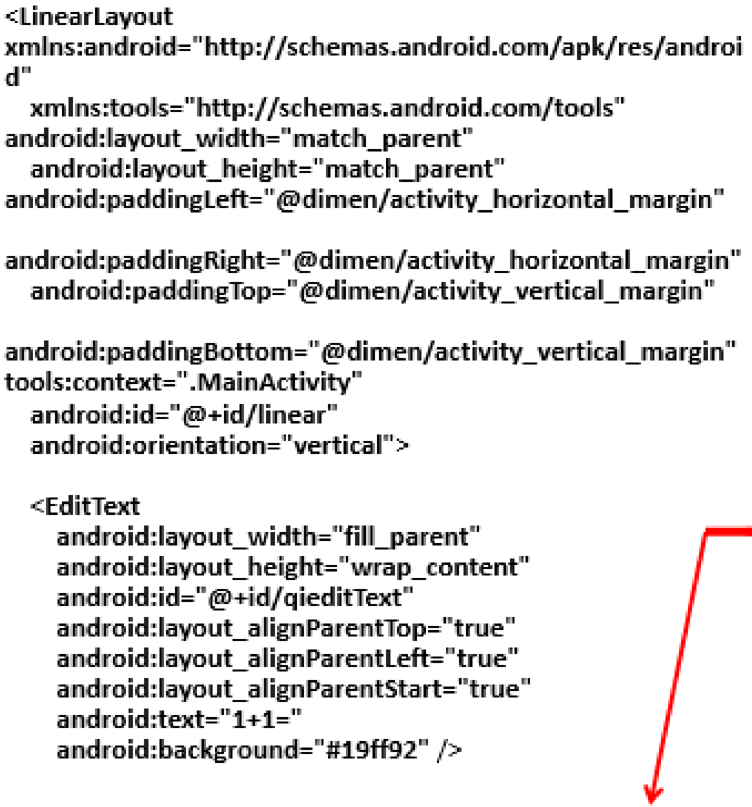
○ 1

○ 2

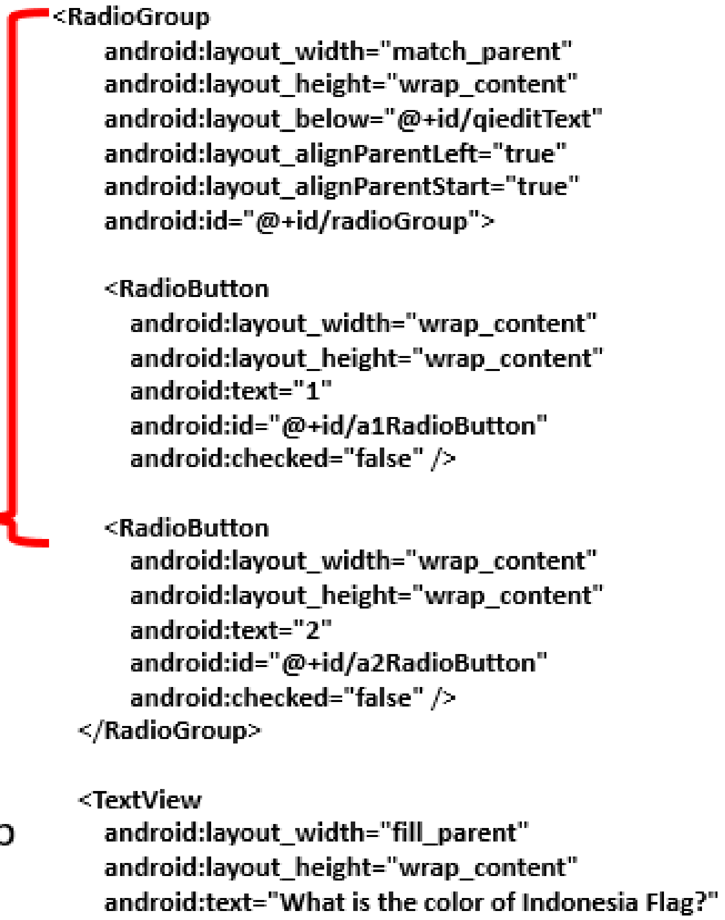
What is the color of Indonesia Flag?

□ red □ White

Two layouts in one design:



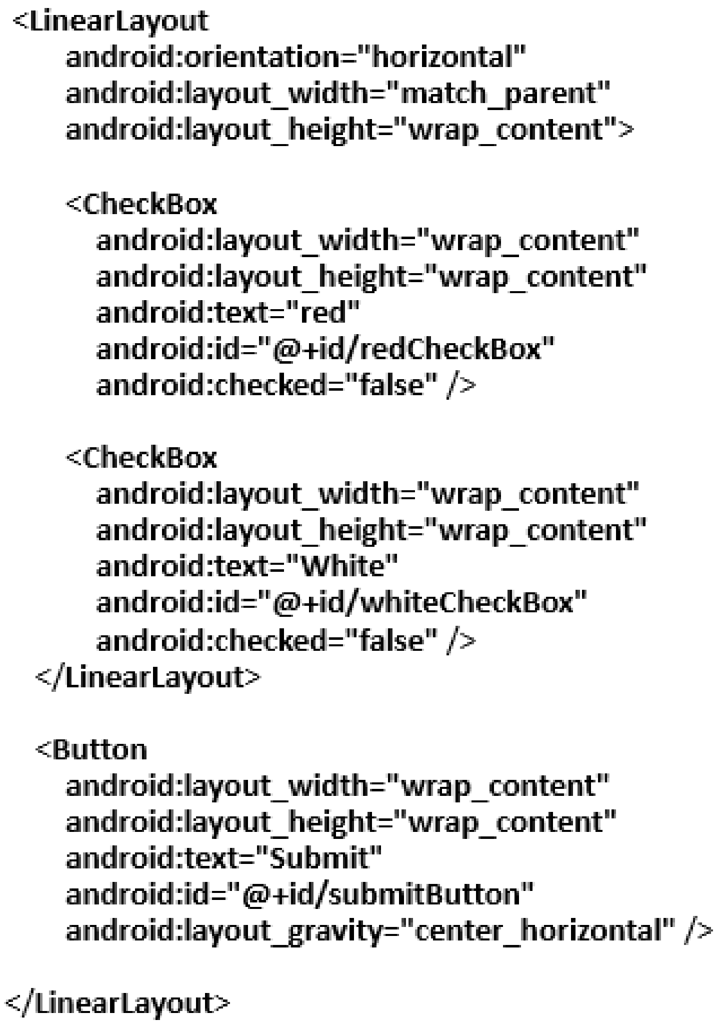
```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/androi
d"
    xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
    android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"

android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity"
    android:id="@+id/linear"
    android:orientation="vertical">

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/qieditText"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:text="1+1="
        android:background="#19ff92" />
```



```
<RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/qieditText"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:id="@+id/radioGroup">

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="1"
            android:id="@+id/a1RadioButton"
            android:checked="false" />

        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="2"
            android:id="@+id/a2RadioButton"
            android:checked="false" />
    </RadioGroup>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="What is the color of Indonesia Flag?"
```

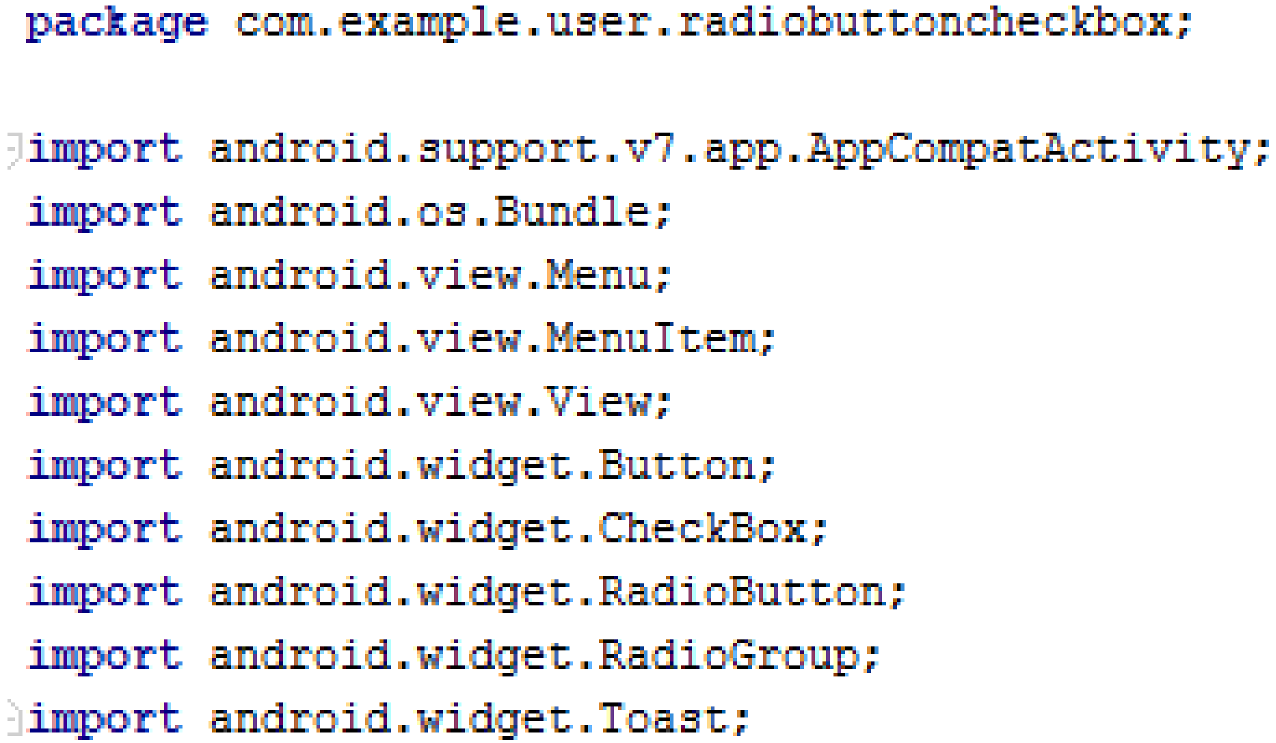
Radio buttons were placed inside of radio Group to make sure user can only select One item

Two layouts in one design:



```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <CheckBox
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="red"
      android:id="@+id/redCheckBox"
      android:checked="false" />

    <CheckBox
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="White"
      android:id="@+id/whiteCheckBox"
      android:checked="false" />
  </LinearLayout>

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Submit"
    android:id="@+id/submitButton"
    android:layout_gravity="center_horizontal" />

</LinearLayout>
```

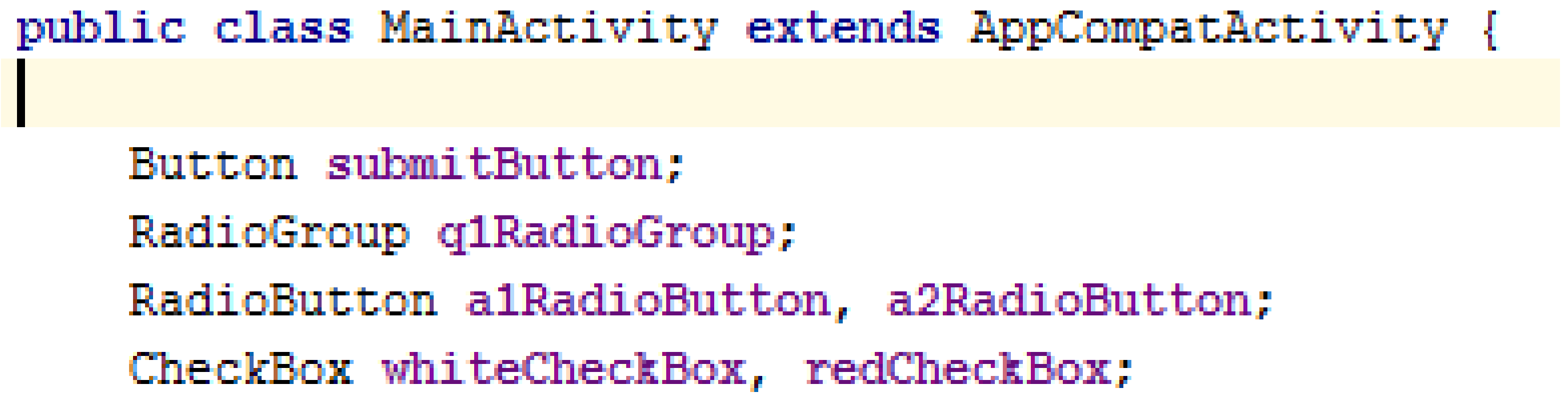
Import libraries



```
package com.example.user.radiobuttoncheckbox;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;
```

- Automatically generated once programmer type the widgets name during declaration
- Instead of ActionBarActivity API version 23 will import AppCompatActivity

Object declaration



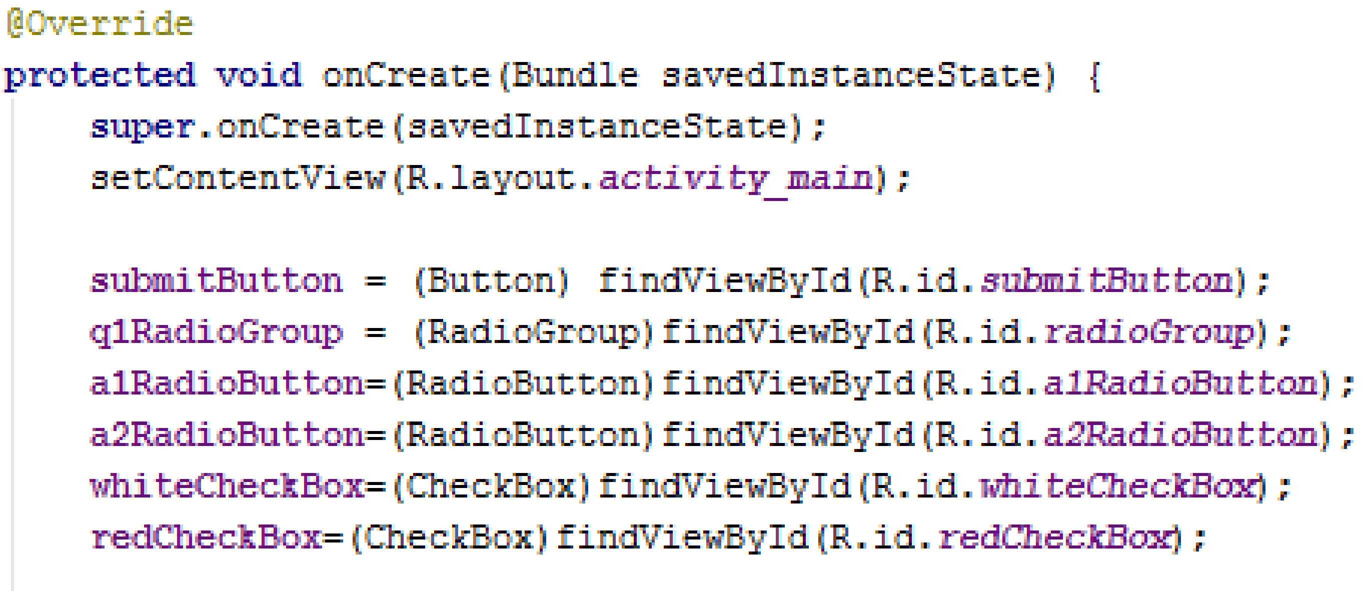
```
public class MainActivity extends AppCompatActivity {

    Button submitButton;
    RadioGroup q1RadioGroup;
    RadioButton a1RadioButton, a2RadioButton;
    CheckBox whiteCheckBox, redCheckBox;
```

Not all widgets have to be declared. Depends on your apps functionality.

If certain value of the widgets will be change during the life cycle of the apps, then you have to declare.

If the value is static (not change) then no need to put inside of the java file.

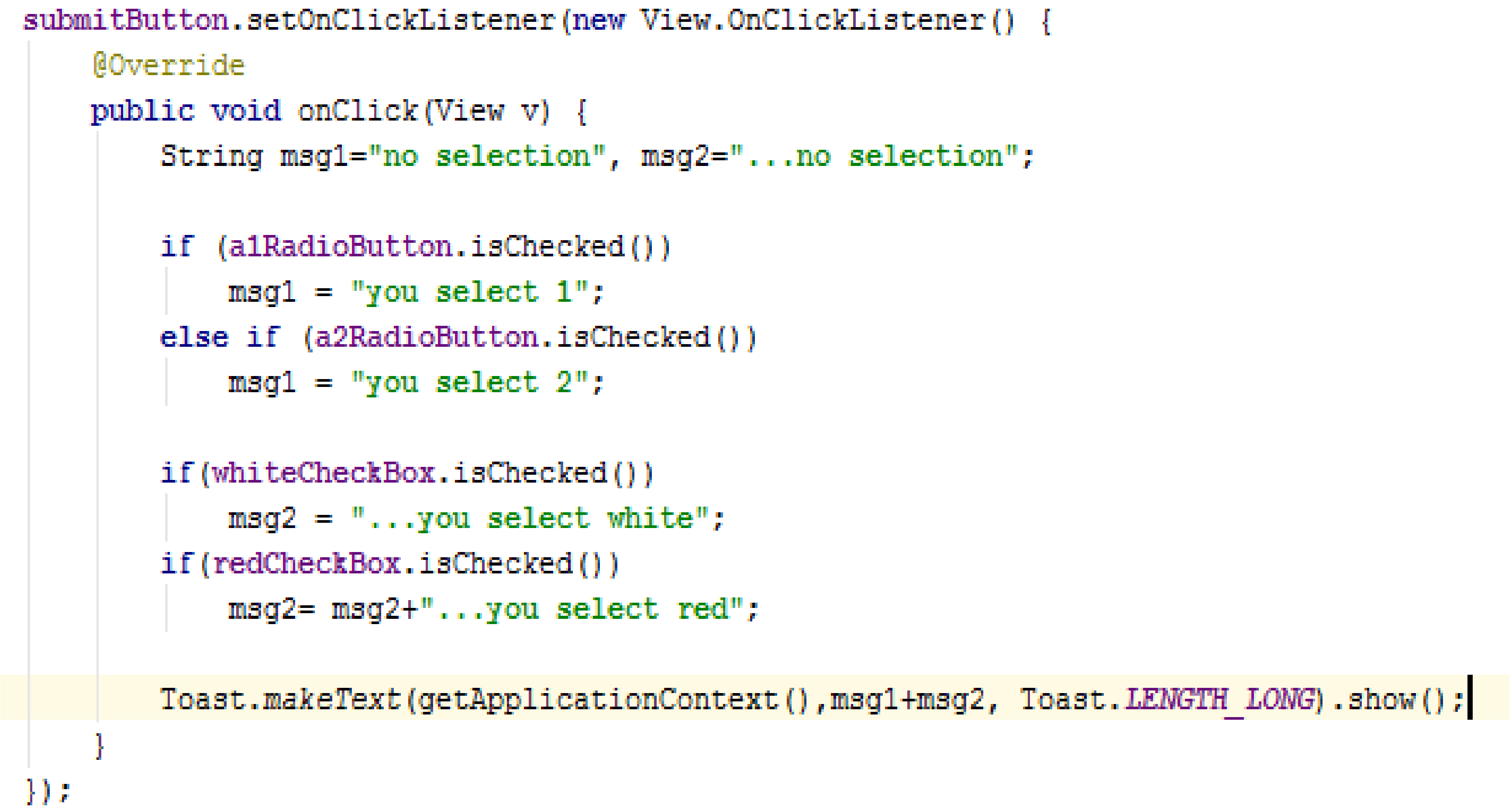
Link Java Objects with widgets



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

    submitButton = (Button) findViewById(R.id.submitButton);
    q1RadioGroup = (RadioGroup)findViewById(R.id.radioGroup);
    a1RadioButton=(RadioButton)findViewById(R.id.a1RadioButton);
    a2RadioButton=(RadioButton)findViewById(R.id.a2RadioButton);
    whiteCheckBox=(CheckBox)findViewById(R.id.whiteCheckBox);
    redCheckBox=(CheckBox)findViewById(R.id.redCheckBox);
```

- It is important for you to change the id of each view (layout/widgets/etc..) inside the xml file corresponding for the GUI design.
- This id will be used to link between the widgets in xml file with the widget objects declared inside of java file.

Program your radio button &check box:



```
submitButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String msg1="no selection", msg2="...no selection";

        if (a1RadioButton.isChecked())
            msg1 = "you select 1";
        else if (a2RadioButton.isChecked())
            msg1 = "you select 2";

        if(whiteCheckBox.isChecked())
            msg2 = "...you select white";
        if(redCheckBox.isChecked())
            msg2= msg2+"...you select red";

        Toast.makeText(getApplicationContext(),msg1+msg2, Toast.LENGTH_LONG).show();
    }
});
```