Mobile Programming

# Application Menu

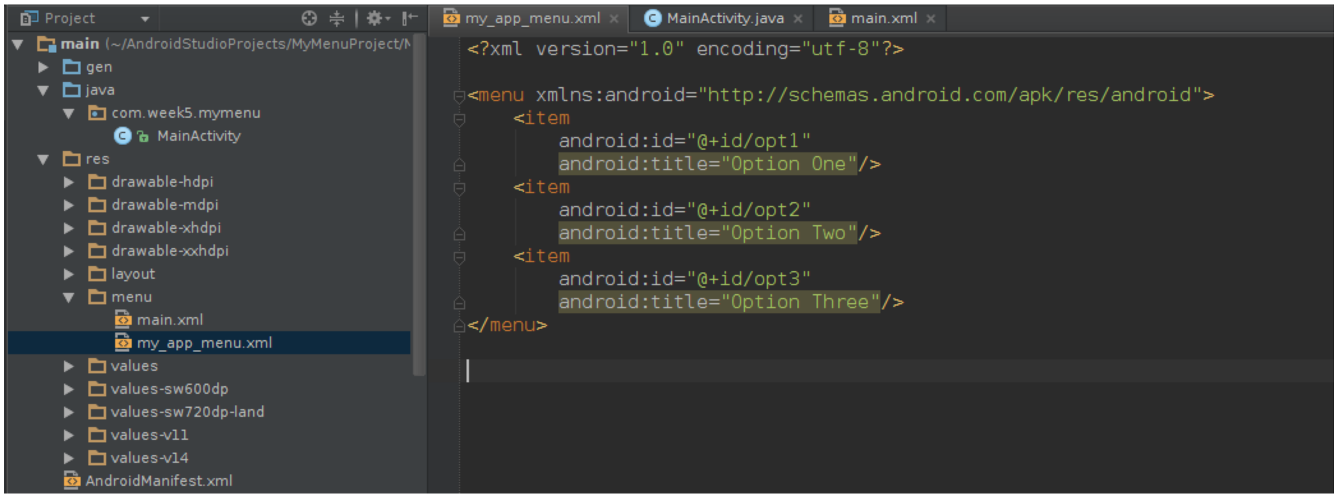
Week #5 [ 19-25 Oct. 2014]

**Topics:**

Options Menu
Floating Contextual Menu
Contextual Action Mode Menu
Popup Menu

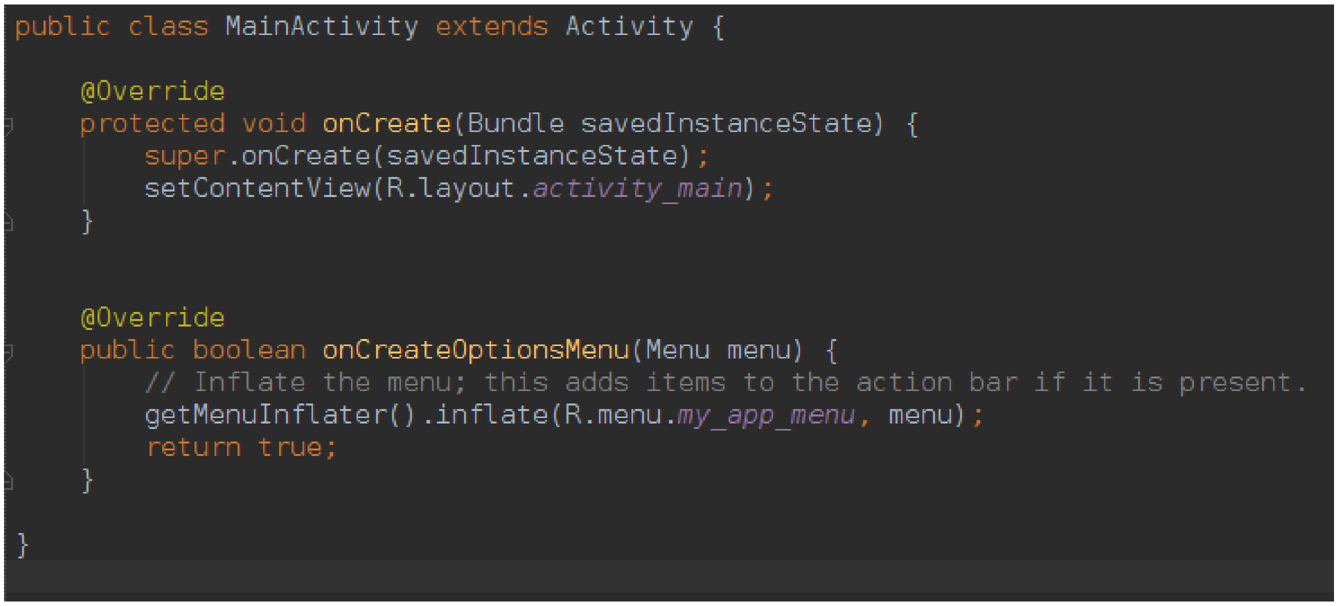
# Options Menu

## 1. Create my_app_menu.xml under res/menu



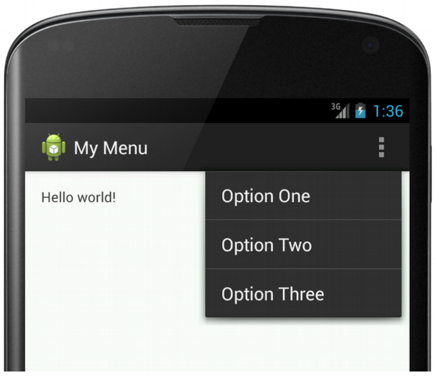
```xml
<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/opt1"
        android:title="Option One"/>
    <item
        android:id="@+id/opt2"
        android:title="Option Two"/>
    <item
        android:id="@+id/opt3"
        android:title="Option Three"/>
</menu>
```

## 2. Override onCreateOptionsMenu() at activity class to inflate menu.xml

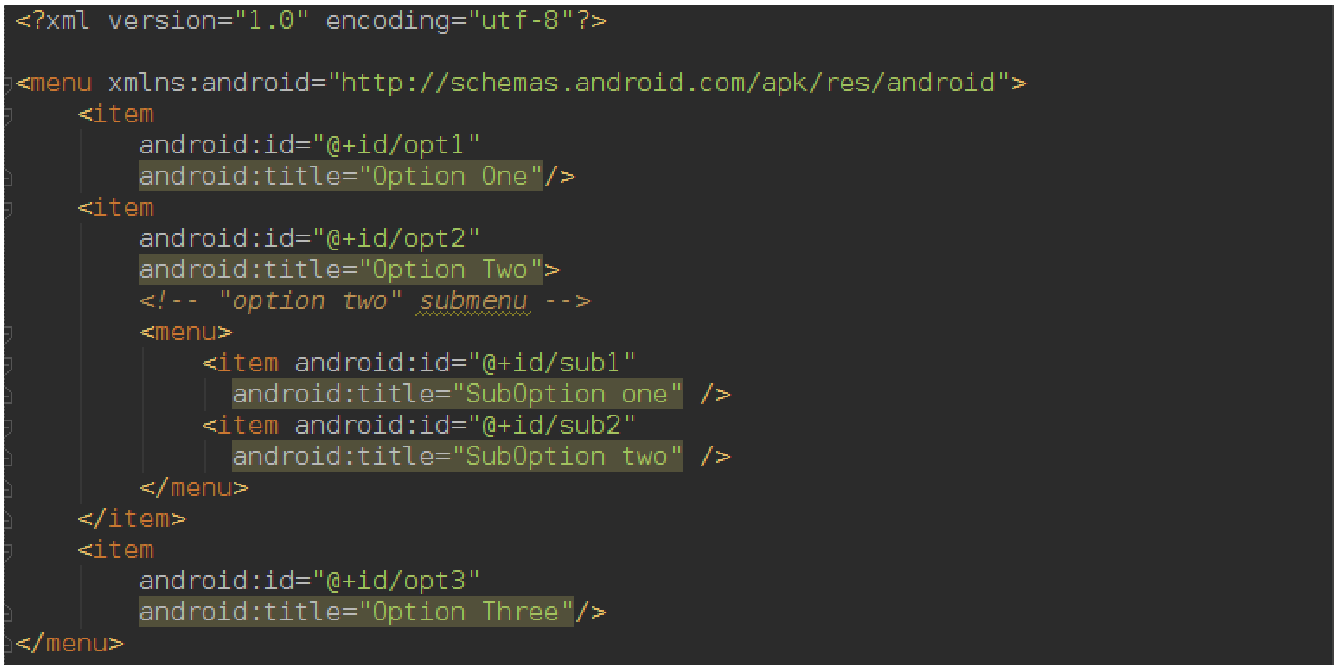


```java
public class MainActivity extends Activity {

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


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

}
```

## 3. Run your projects

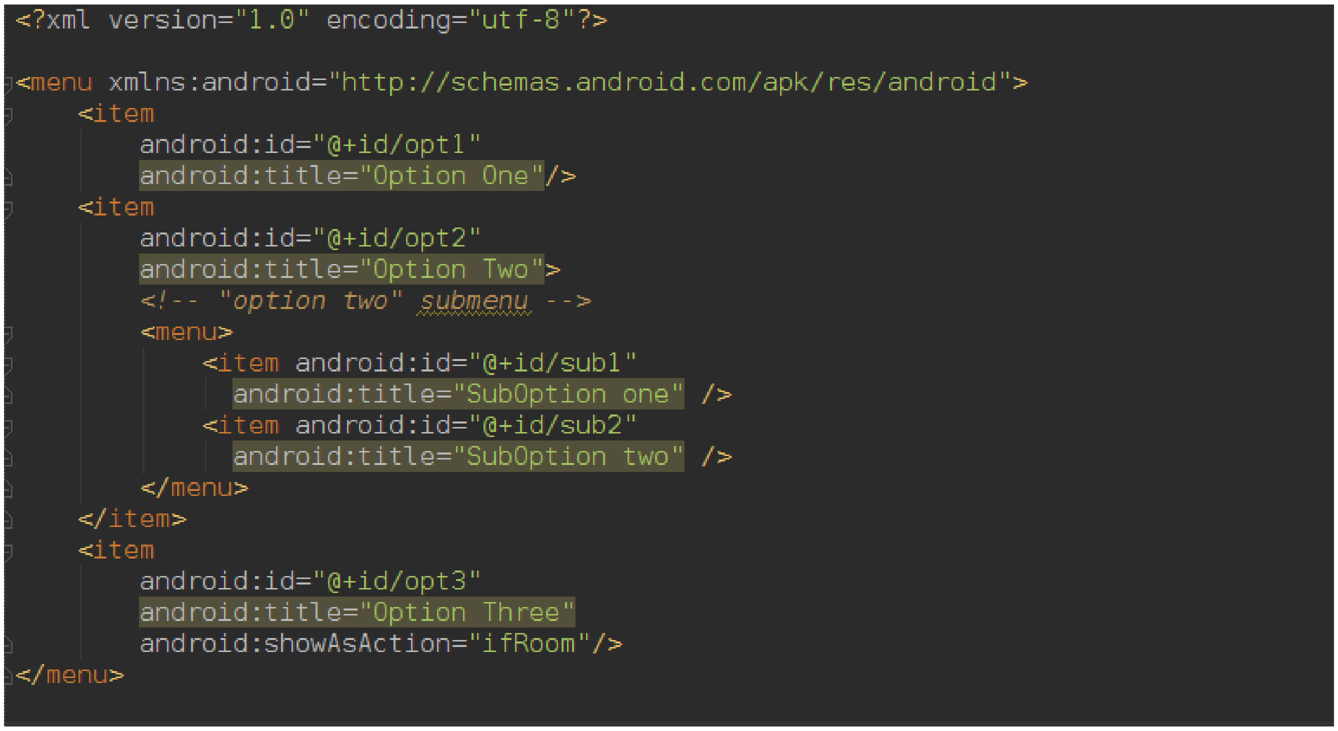


4. Modify my_app_menu.xml



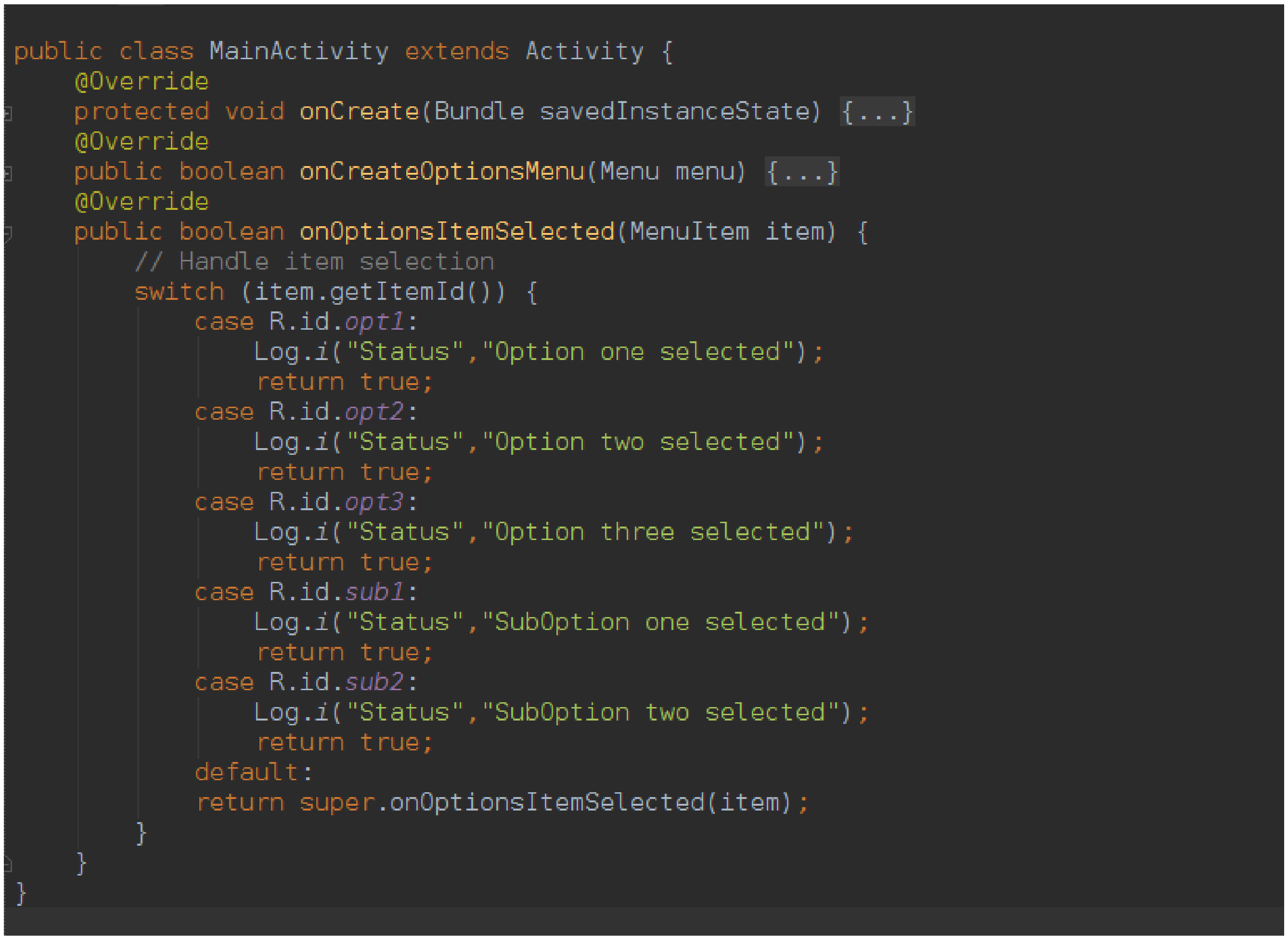
```xml
<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/opt1"
        android:title="Option One"/>
    <item
        android:id="@+id/opt2"
        android:title="Option Two">
        <!-- "option two" submenu -->
        <menu>
            <item android:id="@+id/sub1"
                android:title="SubOption one" />
            <item android:id="@+id/sub2"
                android:title="SubOption two" />
        </menu>
    </item>
    <item
        android:id="@+id/opt3"
        android:title="Option Three"/>
</menu>
```

5. Run and see the changes
6. Modify my_app_menu.xml



```xml
<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/opt1"
        android:title="Option One"/>
    <item
        android:id="@+id/opt2"
        android:title="Option Two">
        <!-- "option two" submenu -->
        <menu>
            <item android:id="@+id/sub1"
                android:title="SubOption one" />
            <item android:id="@+id/sub2"
                android:title="SubOption two" />
        </menu>
    </item>
    <item
        android:id="@+id/opt3"
        android:title="Option Three"
        android:showAsAction="ifRoom"/>
</menu>
```

7. Override onOptionsItemSelected() at activity class to handle click event



```java
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {...}
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {...}
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle item selection
        switch (item.getItemId()) {
            case R.id.opt1:
                Log.i("Status","Option one selected");
                return true;
            case R.id.opt2:
                Log.i("Status","Option two selected");
                return true;
            case R.id.opt3:
                Log.i("Status","Option three selected");
                return true;
            case R.id.sub1:
                Log.i("Status","SubOption one selected");
                return true;
            case R.id.sub2:
                Log.i("Status","SubOption two selected");
                return true;
            default:
            return super.onOptionsItemSelected(item);
        }
    }
}
```

## 8. Run and see LogCat when you press menu item

---

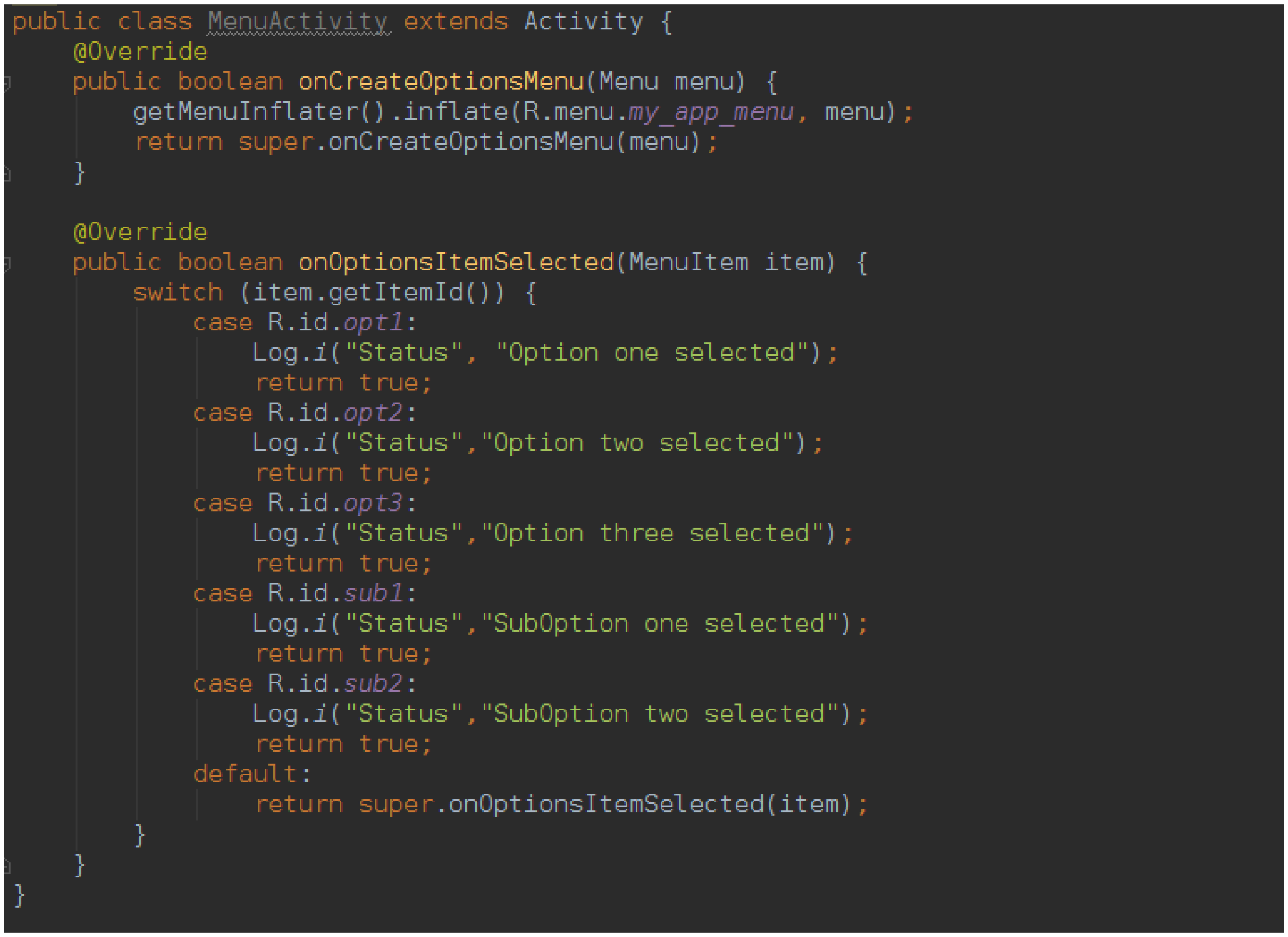
## Evaluation:

1. Where is the location of  the xml file which define menu?
2. What method at activity class should be overrode if you want to inflate menu?
3. What method at activity class should be overrode if you want to handle click on menu?
4. If you want to make some item show at action bar, what attribut that you should gave?
5. How to create subitem?
6. What is the argument of onItemOptionSelected() method?

---

## Option Menu For All Activity

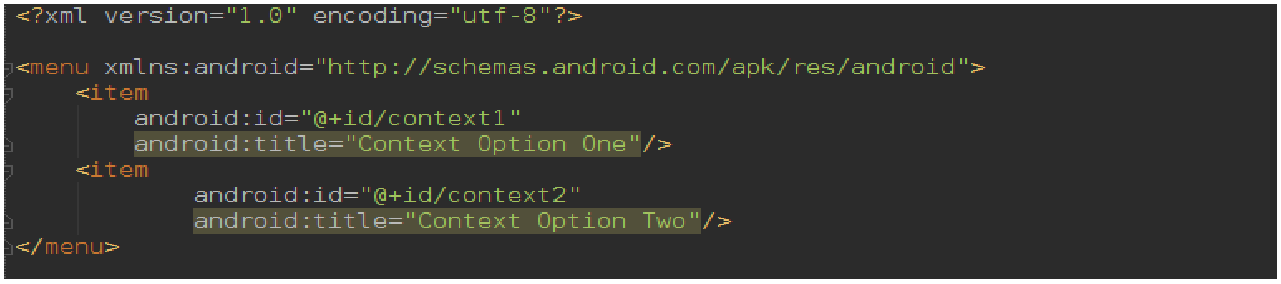
To provide same option menu to all activity in your application, you can make activity class that implements nothing except onCreateOptionsMenu() and onOptionsItemSelected() as a superclass of all the activity in your application.



```
public class MenuActivity extends Activity {
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.my_app_menu, menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.opt1:
                Log.i("Status", "Option one selected");
                return true;
            case R.id.opt2:
                Log.i("Status","Option two selected");
                return true;
            case R.id.opt3:
                Log.i("Status","Option three selected");
                return true;
            case R.id.sub1:
                Log.i("Status","SubOption one selected");
                return true;
            case R.id.sub2:
                Log.i("Status","SubOption two selected");
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
}
```

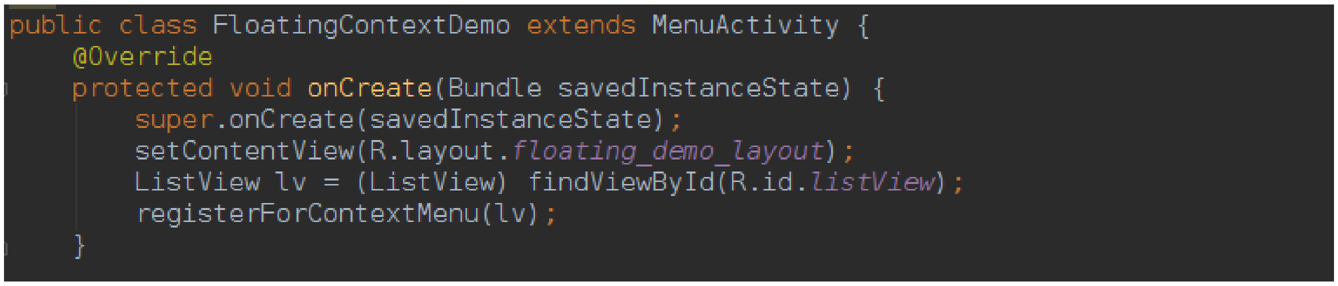
## Floating Contextual Menu

1. Create floating_demo_layout.xml, add ListView to your layout then fill with dummy data (see previous worksheet to learn how to do that)
2. Create FloatingContextDemo.java activity and use floating_demo_layout.xml as layout file
3. Create float_context_menu.xml



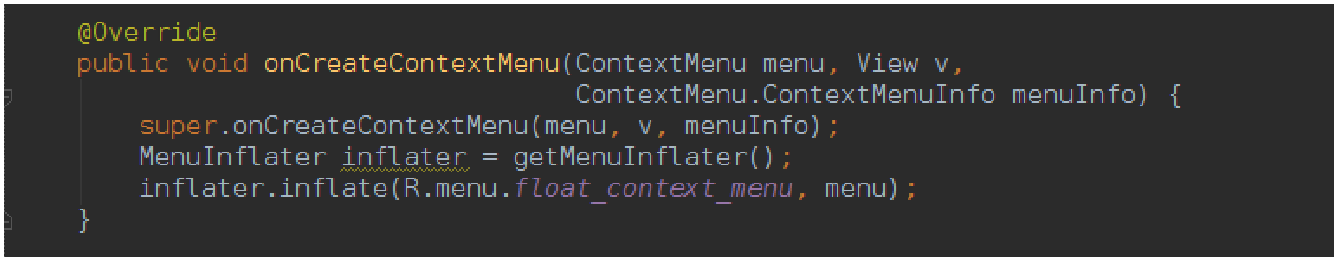
```xml
<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/context1"
        android:title="Context Option One"/>
    <item
            android:id="@+id/context2"
            android:title="Context Option Two"/>
</menu>
```

## 4. Register context menu for ListView



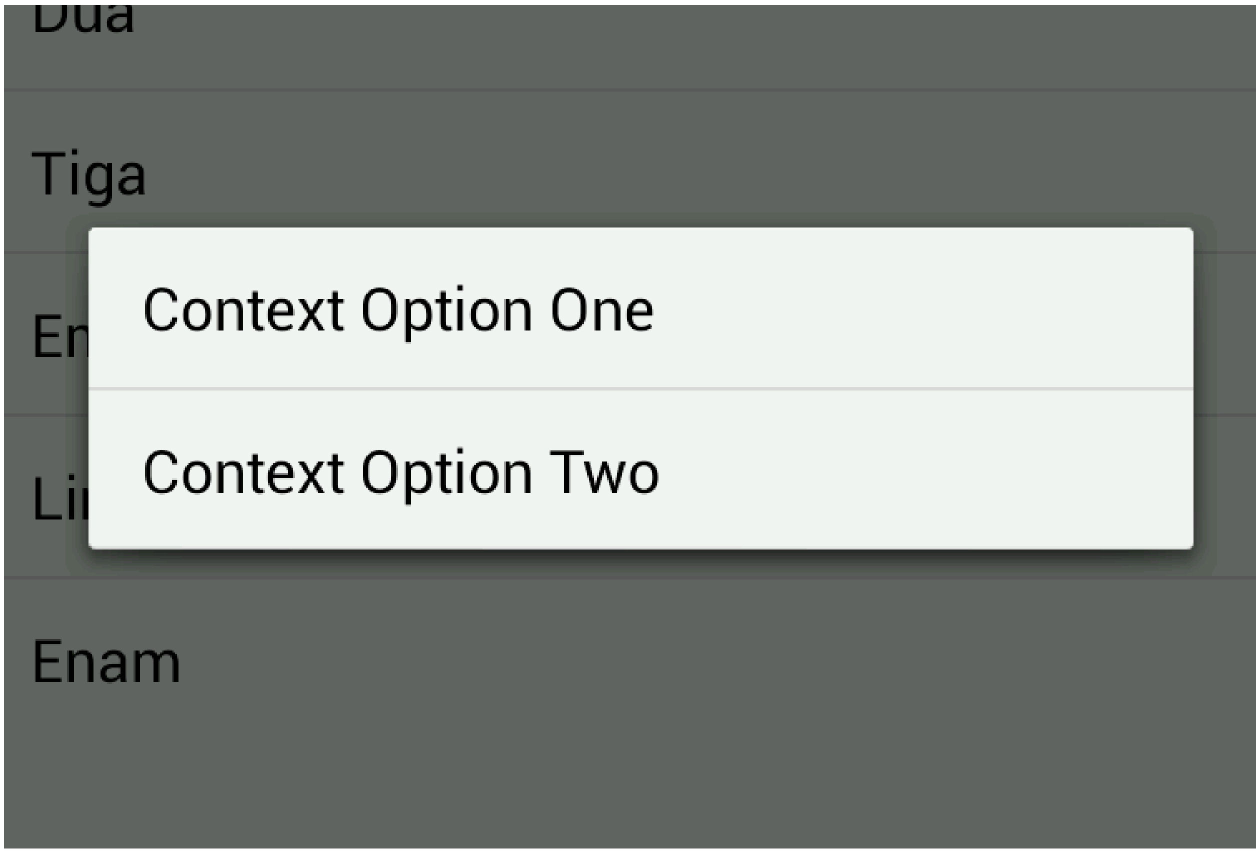
```java
public class FloatingContextDemo extends MenuActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.floating_demo_layout);
        ListView lv = (ListView) findViewById(R.id.listView);
        registerForContextMenu(lv);
    }
```

## 5. Implement onCreateContextMenu() method in your Activity

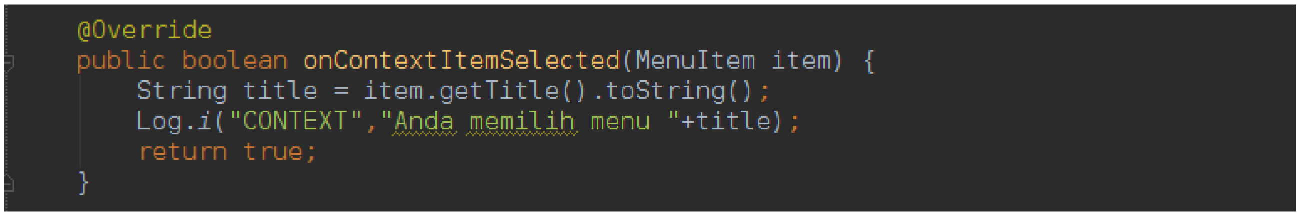


```java
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
                                    ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.float_context_menu, menu);
    }
```

## 6. Run and see the result when you long pres list item



7. Implement onContextItemSelected() to handle click

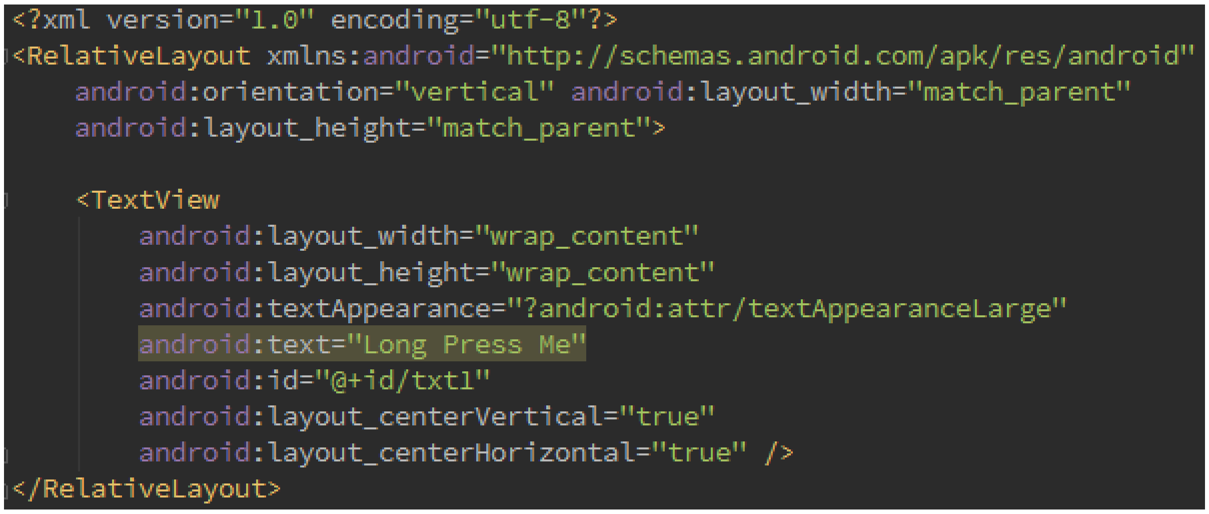


```
@Override
public boolean onContextItemSelected(MenuItem item) {
    String title = item.getTitle().toString();
    Log.i("CONTEXT","Anda memilih menu "+title);
    return true;
}
```

8. Run and see the result

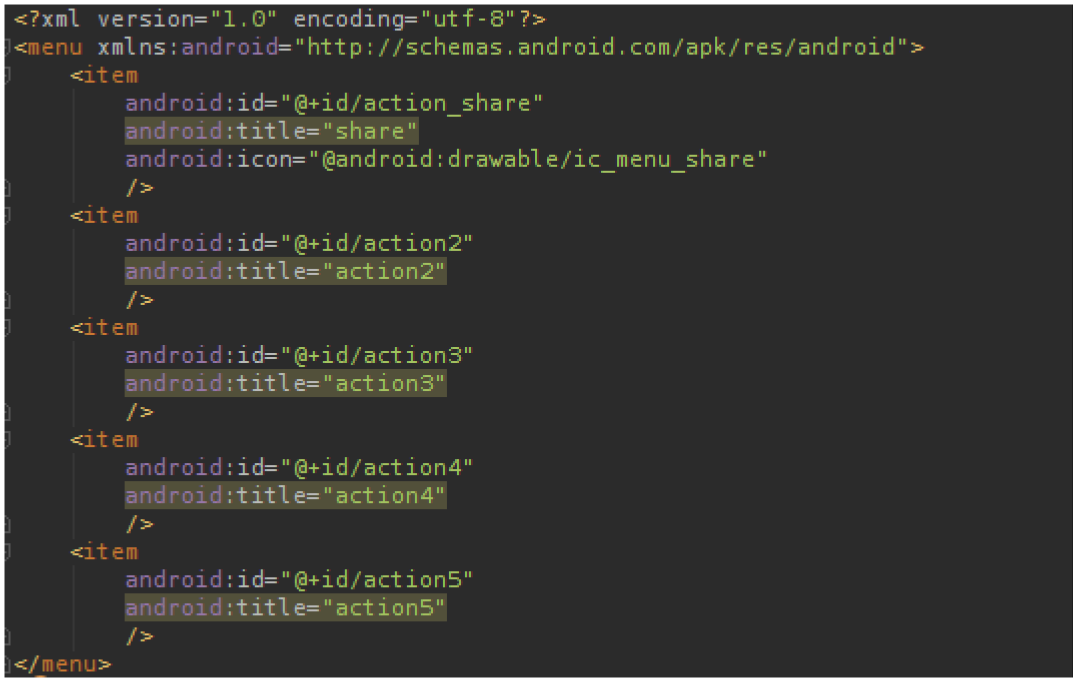
## Contextual Action Mode Menu

1. If you want to invoke the contextual action mode only when the user selects specific views, you should:
   1.1. Implement the ActionMode.Callback interface. In its callback methods, you can specify the actions for the contextual action bar, respond to click events on action items, and handle other lifecycle events for the action mode.
   1.2. Call startActionMode() when you want to show the bar (such as when the user long-clicks the view).
   1.3. Example:
      a. Create simple layout with TextView



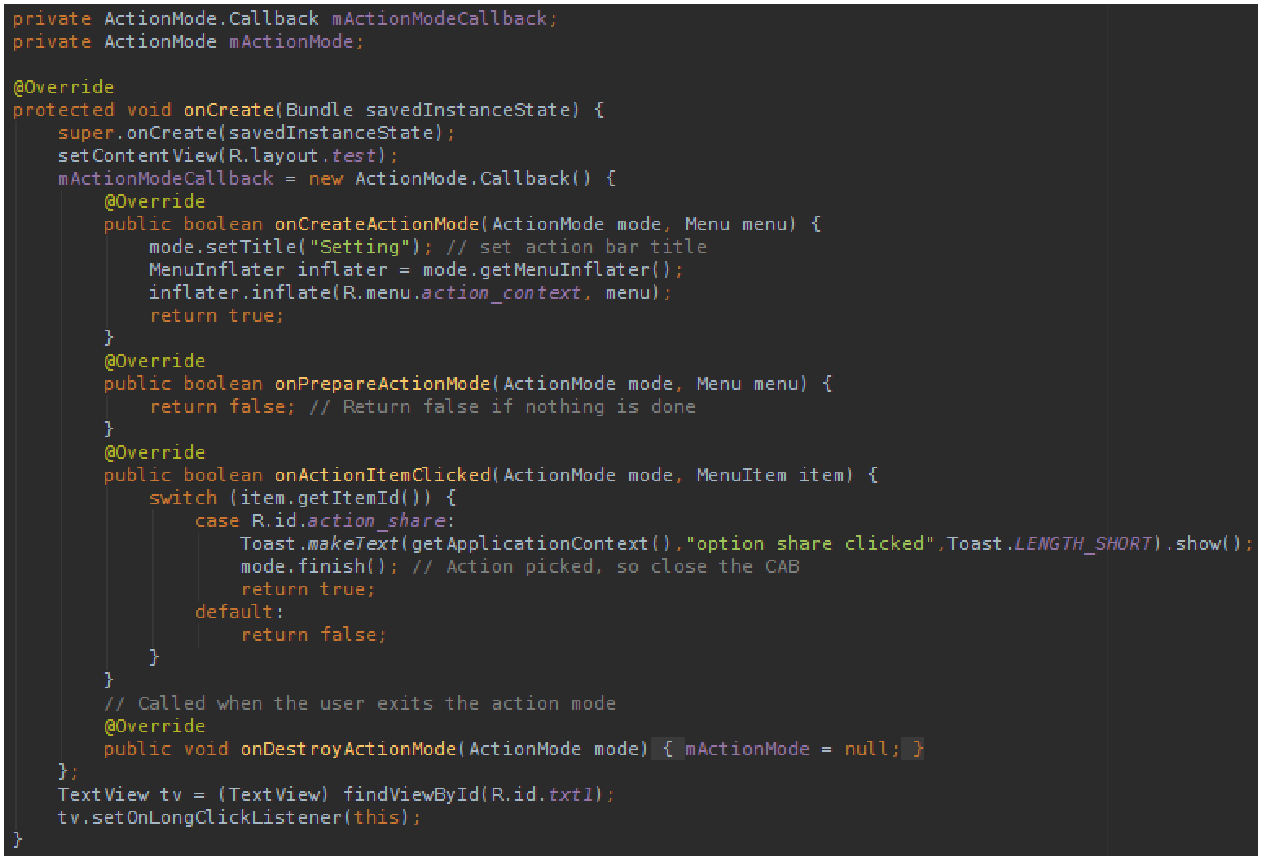
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Long Press Me"
        android:id="@+id/txt1"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

      b. Create menu file to define ActionMode menu. (for this example: res/menu/action_context.xml)



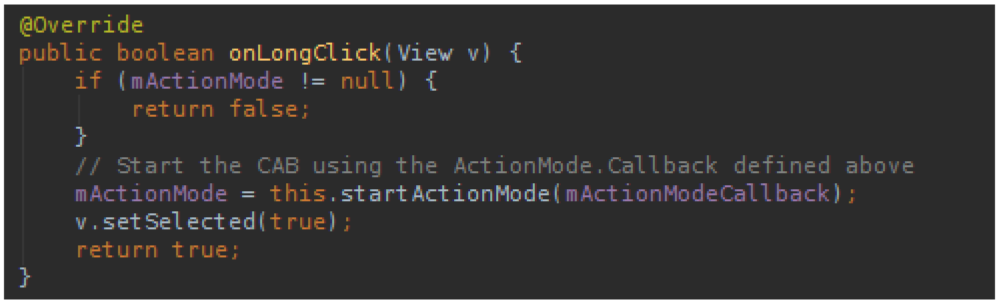
```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/action_share"
        android:title="share"
        android:icon="@android:drawable/ic_menu_share"
        />
    <item
        android:id="@+id/action2"
        android:title="action2"
        />
    <item
        android:id="@+id/action3"
        android:title="action3"
        />
    <item
        android:id="@+id/action4"
        android:title="action4"
        />
    <item
        android:id="@+id/action5"
        android:title="action5"
        />
</menu>
```

c. Create property that implement ActionMode.callback interface at Activity class



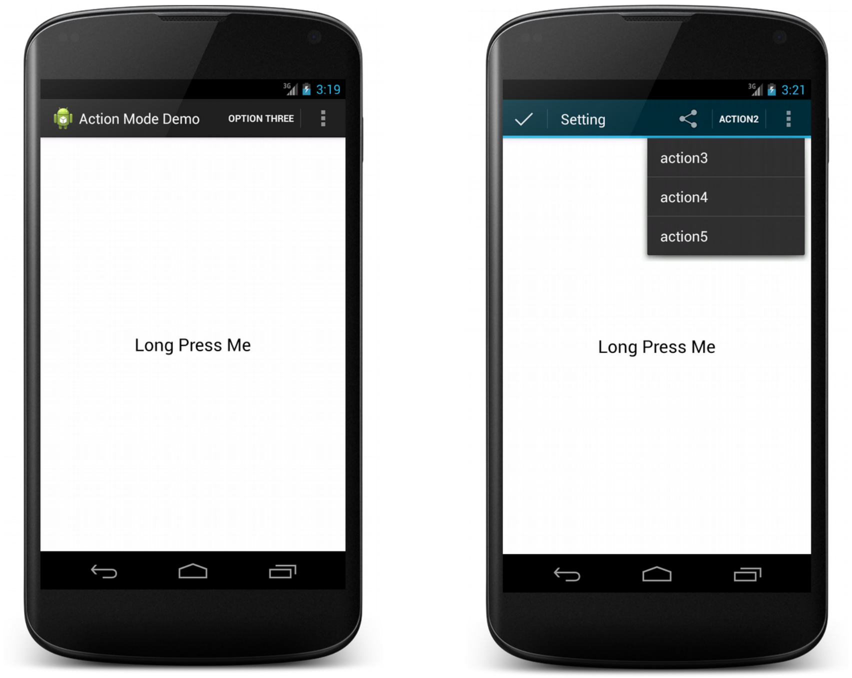
```java
private ActionMode.Callback mActionModeCallback;
private ActionMode mActionMode;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.test);
    mActionModeCallback = new ActionMode.Callback() {
        @Override
        public boolean onCreateActionMode(ActionMode mode, Menu menu) {
            mode.setTitle("Setting"); // set action bar title
            MenuInflater inflater = mode.getMenuInflater();
            inflater.inflate(R.menu.action_context, menu);
            return true;
        }
        @Override
        public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
            return false; // Return false if nothing is done
        }
        @Override
        public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
            switch (item.getItemId()) {
                case R.id.action_share:
                    Toast.makeText(getApplicationContext(),"option share clicked",Toast.LENGTH_SHORT).show();
                    mode.finish(); // Action picked, so close the CAB
                    return true;
                default:
                    return false;
            }
        }
        // Called when the user exits the action mode
        @Override
        public void onDestroyActionMode(ActionMode mode) { mActionMode = null; }
    };
    TextView tv = (TextView) findViewById(R.id.txt1);
    tv.setOnLongClickListener(this);
}
```

d.Call startActionMode() when long-click on TextView

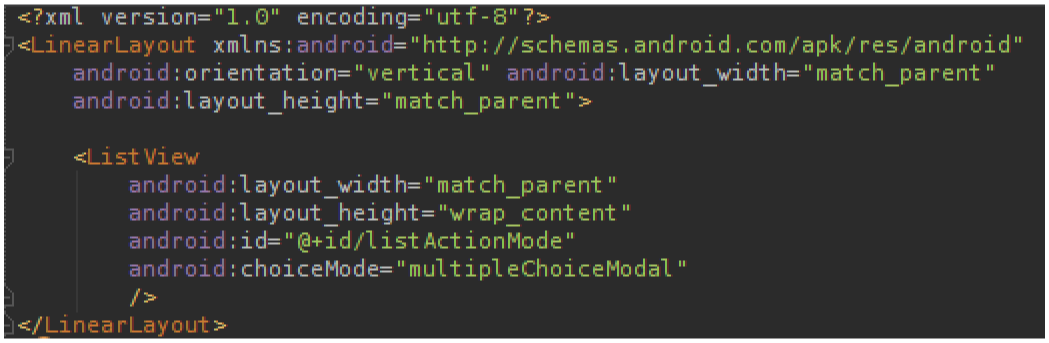


```
@Override
public boolean onLongClick(View v) {
    if (mActionMode != null) {
        return false;
    }
    // Start the CAB using the ActionMode.Callback defined above
    mActionMode = this.startActionMode(mActionModeCallback);
    v.setSelected(true);
    return true;
}
```

e. Run and see the result



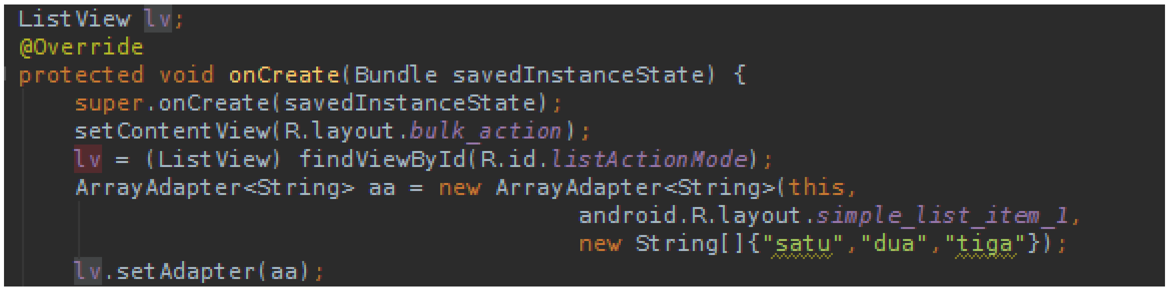
2. Invoke contextual action mode when the user selects specific item or multiple item
    a. Create ListView with multiple choice modal



```
<?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">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listActionMode"
        android:choiceMode="multipleChoiceModal"
        />
</LinearLayout>
```

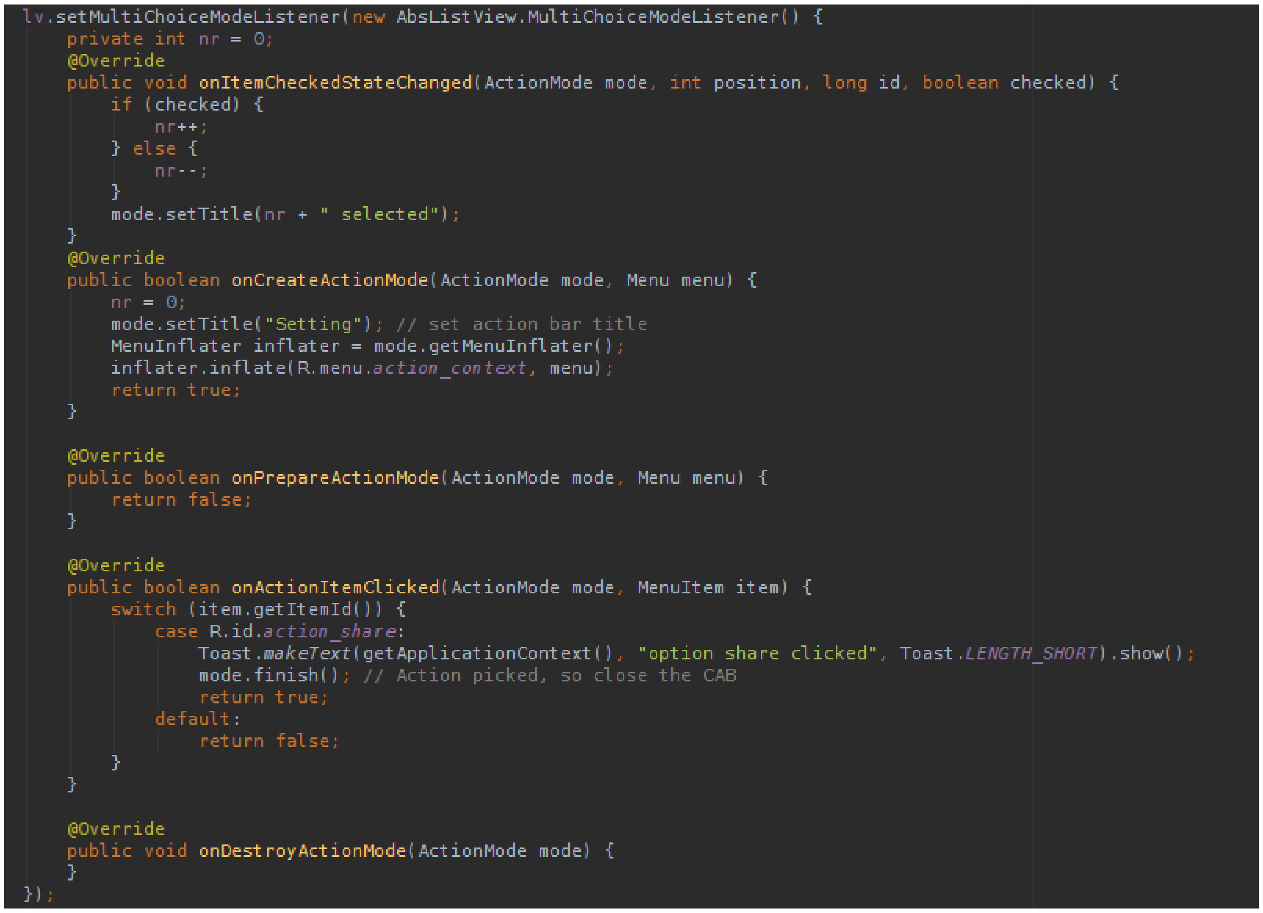
    b. Create menu file to define ActionMode menu. For this example, use previous file (action_context.xml)

## c. Fill listView with dummy data



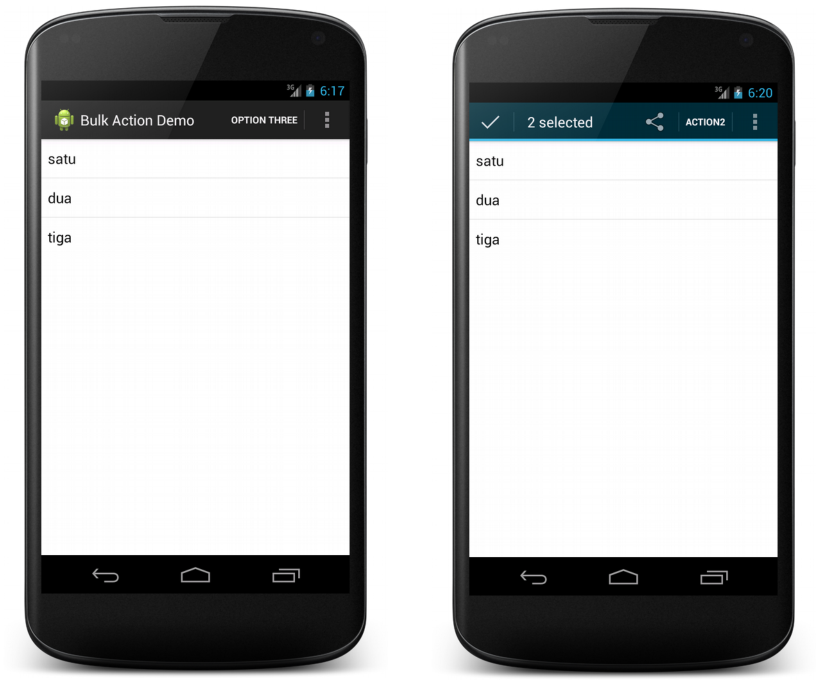
```
ListView lv;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.bulk_action);
    lv = (ListView) findViewById(R.id.listActionMode);
    ArrayAdapter<String> aa = new ArrayAdapter<String>(this,
                                    android.R.layout.simple_list_item_1,
                                    new String[]{"satu","dua","tiga"});
    lv.setAdapter(aa);
```

## d. Set MultiChoiceModeListener for listview to enable contextual action mode



```
lv.setMultiChoiceModeListener(new AbsListView.MultiChoiceModeListener() {
    private int nr = 0;
    @Override
    public void onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) {
        if (checked) {
            nr++;
        } else {
            nr--;
        }
        mode.setTitle(nr + " selected");
    }
    @Override
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {
        nr = 0;
        mode.setTitle("Setting"); // set action bar title
        MenuInflater inflater = mode.getMenuInflater();
        inflater.inflate(R.menu.action_context, menu);
        return true;
    }

    @Override
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
        return false;
    }

    @Override
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
        switch (item.getItemId()) {
            case R.id.action_share:
                Toast.makeText(getApplicationContext(), "option share clicked", Toast.LENGTH_SHORT).show();
                mode.finish(); // Action picked, so close the CAB
                return true;
            default:
                return false;
        }
    }

    @Override
    public void onDestroyActionMode(ActionMode mode) {
    }
});
```

## e. Run and see the result



---

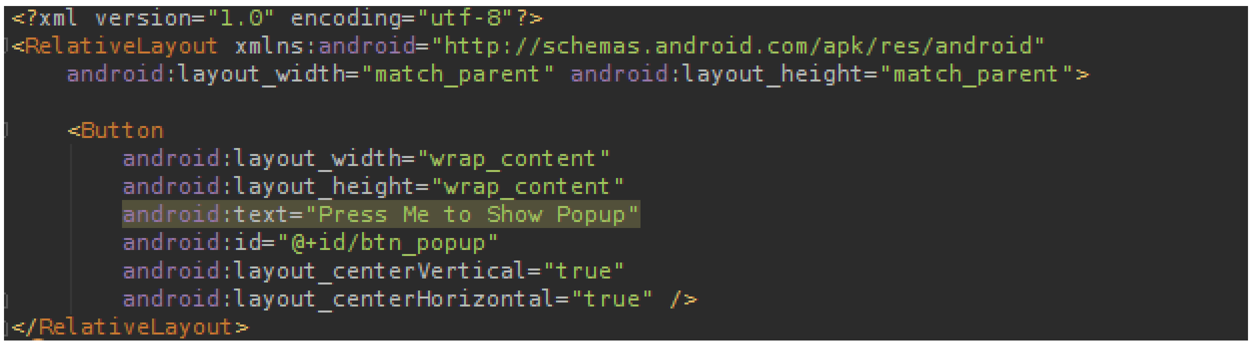
## Evaluation:

1. How to define RelativeLayout?
2. What attribute android:align_ParentRight=”true” means?
3. Please capture the screen of your works

---

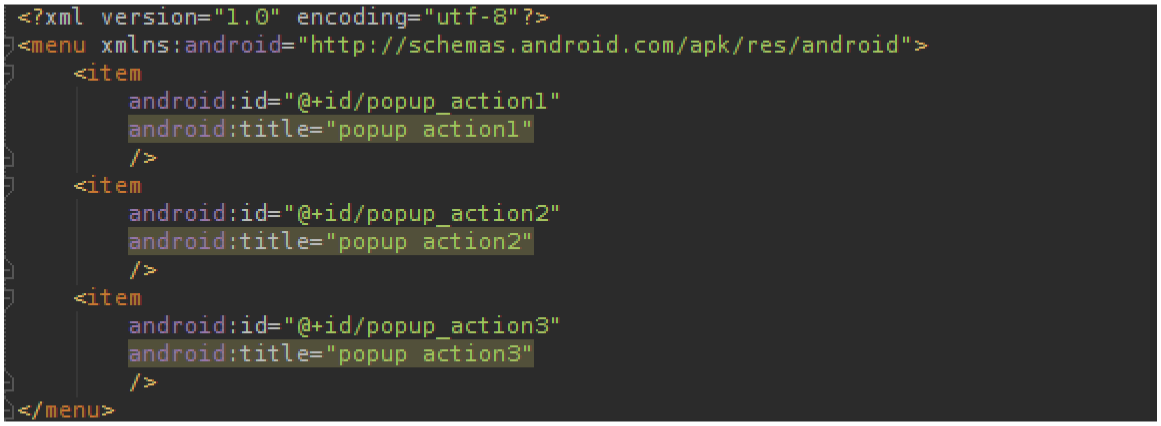
## POPUP Menu

### 1. Create Layout File



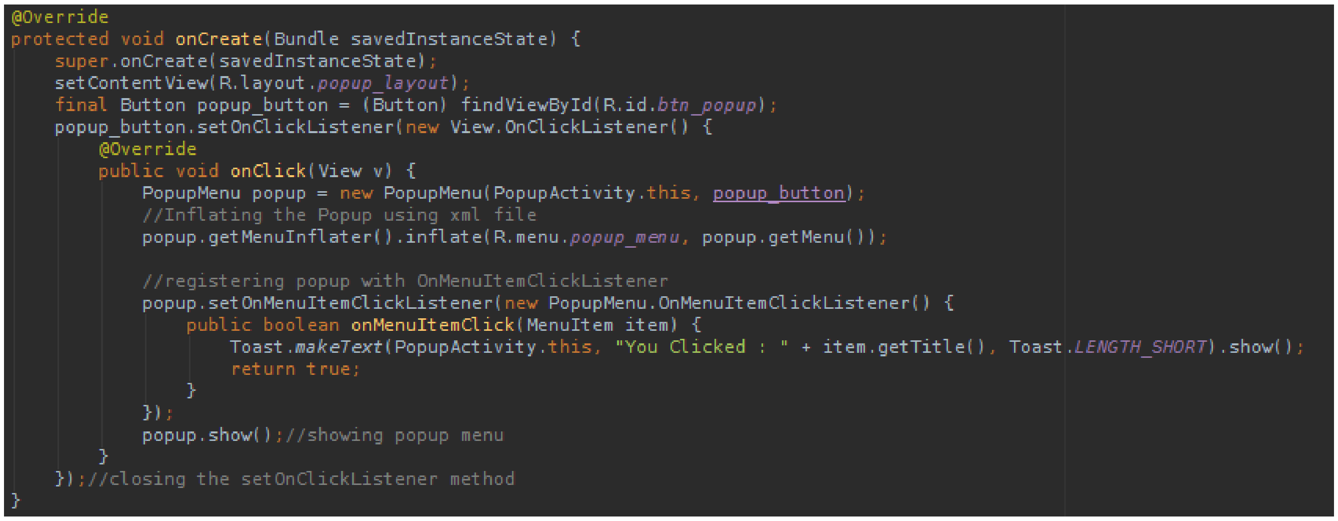
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Press Me to Show Popup"
        android:id="@+id/btn_popup"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

### 2. Create menu file to define popup menu item



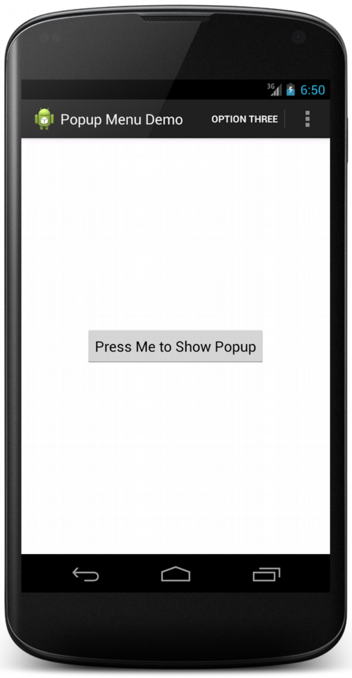
```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/popup_action1"
        android:title="popup action1"
        />
    <item
        android:id="@+id/popup_action2"
        android:title="popup action2"
        />
    <item
        android:id="@+id/popup_action3"
        android:title="popup action3"
        />
</menu>
```

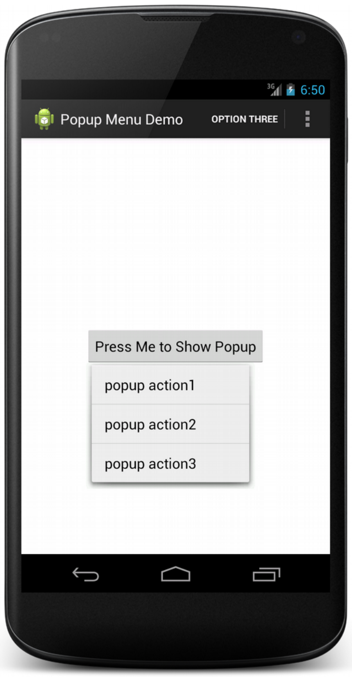
## 3. Set onClickListener to show popup



```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.popup_layout);
    final Button popup_button = (Button) findViewById(R.id.btn_popup);
    popup_button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            PopupMenu popup = new PopupMenu(PopupActivity.this, popup_button);
            //Inflating the Popup using xml file
            popup.getMenuInflater().inflate(R.menu.popup_menu, popup.getMenu());

            //registering popup with OnMenuItemClickListener
            popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
                public boolean onMenuItemClick(MenuItem item) {
                    Toast.makeText(PopupActivity.this, "You Clicked : " + item.getTitle(), Toast.LENGTH_SHORT).show();
                    return true;
                }
            });
            popup.show();//showing popup menu
        }
    });//closing the setOnClickListener method
}
```

## 4. Run and see the result


6:50
Popup Menu Demo
OPTION THREE
Press Me to Show Popup


6:50
Popup Menu Demo
OPTION THREE
Press Me to Show Popup
popup action1
popup action2
popup action3