ITMD 455/555 *Intelligent Device Applications* Lab 2

Temperature Converter (part deux) App- 50 points

Introduction. This lab continues with lab 1 logic but throws in a few more goodies namely a seekBar, checkbox, listview with an arrayAdapter and a Viewstub.

The code is well laid out replete with `comments`. Study over the code syntax, the logic (esp. the objects used), the import statements of the Java and XML files VERY carefully, line by line if necessary as that's one way to learn big time!

STEP 1 Creating a New Android Project

Create a new project called with an Empty Activity when prompted. When configuring your project name it **TempConverter2**. Package name can be **com.example.tempconverter2**. Save your project to a desired location. Language choice will be **Java**. Choose an appropriate API like API **26** (Android 8 Oreo). Click Finish when complete.

STEP 2 Completely override your `activity_main.xml` file as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:context="com.example.mypackage.tempconverter2.MainActivity">

    <TextView android:id="@+id/textview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <!-- add View as gap for 4 blank lines -->
    <View
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:id="@+id/view">
    </View>

    <SeekBar android:id="@+id/seekbar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:max="100"
        android:minWidth="250dp"
        android:layout_below="@+id/view"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
    <CheckBox
        android:id="@+id/checkBox1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show 5 Day forecast"
        android:layout_below="@+id/seekbar"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="49dp" />

    <ViewStub
        android:id="@+id/viewStub1"
        android:layout="@layout/stubview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Never mind the warning in red for the **ViewStub** tag, we'll fix in the next step by adding the needed layout!

STEP 3 Add an Android XML file in your layout folder call it `stubview`. Choose a LinearLayout as your Root element.

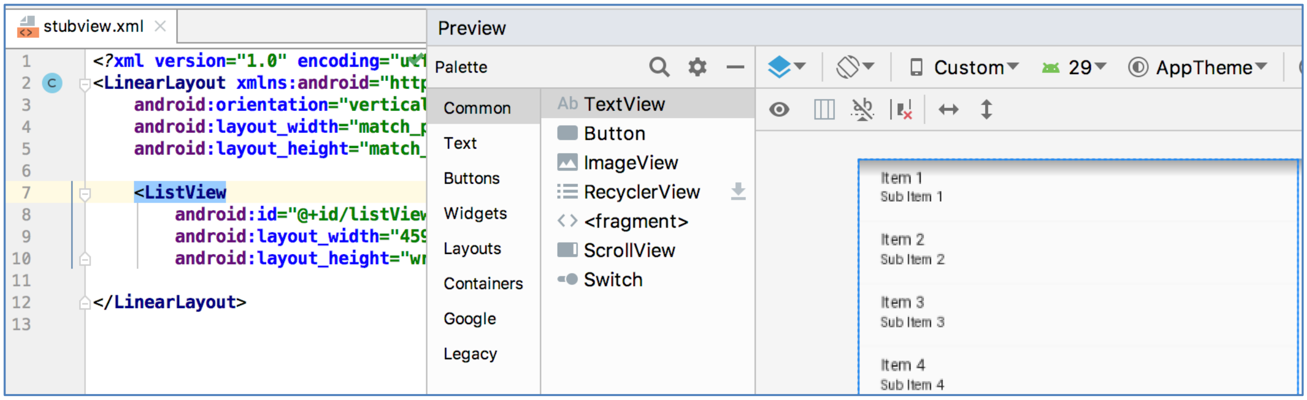
To add in a layout file, right click on your **res/layout** folder and choose New > XML > Layout XML File. Name the file **stubview**. You can put in LinearLayout as your 'Root Tag' if it is not given as a choice. Use the following code for the new file:

```
<?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:id="@+id/listView"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"/>

</LinearLayout>
```

You can now observe in your stubview XML file in design view, a nice ListView layout destined to allow for a series of temperatures to fill each row of the listview. Snapshot follows.



```
<?xml version="1.0" encoding="ut
<LinearLayout xmlns:android="http
    android:orientation="vertical
    android:layout_width="match_p
    android:layout_height="match_

    <ListView
        android:id="@+id/listView
        android:layout_width="459
        android:layout_height="wr

</LinearLayout>
```

STEP 4 Finally add the following code to your `MainActivity` java file

Include the entire code that follows after the following *opening* line of your class declaration, up to but NOT including the closing brace (}) of your *class*:

```
public class MainActivity extends AppCompatActivity {


SeekBar seekBar;  //declare seekbar object
TextView textView;
//declare member variables for SeekBar
int discrete = 0;
int start = 50;
int start_position = 50; //progress tracker
int temp = 0;
//declare objects for ViewStub
ViewStub stub;
CheckBox checkBox;
ListView lv; //declare Listview object

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

    //declare viewstub object
    stub = findViewById(R.id.viewStub1);
    @SuppressWarnings("unused")
    View inflated = stub.inflate();
    stub.setVisibility(View.INVISIBLE);

    //ViewStub logic
    checkBox = findViewById(R.id.checkBox1);

    //handle checkbox click event
    checkBox.setOnCheckedChangeListener(new CheckBox.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton arg0, boolean isChecked) {
    if (isChecked) {
        //remove objs from parent view to allow for child view objs
        checkBox.setVisibility(View.GONE);
        seekBar.setVisibility(View.GONE);
        textView.setVisibility(View.GONE);
        stub.setVisibility(View.VISIBLE);
    }
    }
    });

    //seekbar logic
    textView = findViewById(R.id.textview);
    textView.setText("     Celsius at 0 degrees");
    //set default view
    seekBar = findViewById(R.id.seekbar);
    seekBar.setProgress(start_position);

    //create event handler for SeekBar
    seekBar.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {
    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
         if (temp == 0)  //for initial view result
           Toast.makeText(getBaseContext(), "Fahrenheit result: 32 degrees",
                        Toast.LENGTH_SHORT).show();
         else
           Toast.makeText(getBaseContext(), "Fahrenheit result: "
                            + String.valueOf(discrete) +
                            " degrees",Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {   }

    @Override
    public void onProgressChanged(SeekBar seekBar,
                                 int progress, boolean fromUser) {
       // To convert progress passed as discrete (Fahrenheit) value
       temp = progress - start;
       discrete = (int) Math.round((((temp * 9.0) /
               5.0) + 32)); //convert C to F temp
       textView.setText("     Celsius at " + temp + " degrees");
    }
    });
    //Listview logic
    String[] wkTemps = new String[]{"1", "-10", "0", "30", "10"};

    lv = findViewById(R.id.listView);
    @SuppressWarnings({"unchecked", "rawtypes"})
    /*
     * To use a basic ArrayAdapter, you just need to initialize the adapter and
     * attach the adapter to the ListView. First, initialize the adapter...:
     *
    */
    ArrayAdapter adapter = new ArrayAdapter(this,
      android.R.layout.simple_list_item_1,
      android.R.id.text1, wkTemps);
    // Assign adapter to ListView
    lv.setAdapter(adapter);

}//end onCreate method
```

Add the following import statements as well into your file:

```
import android.view.View;
import android.view.ViewStub;
import android.widget.ArrayAdapter;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.ListView;
import android.widget.SeekBar;
import android.widget.SeekBar.OnSeekBarChangeListener;
import android.widget.TextView;
import android.widget.Toast;
```

STEP 5 Test drive your app

Run your app at this point and it should fly. Test first your SeekBar and see if you get the results your supposed to via the Toast message. Then click on the CheckBox and you should see your Listview result.

STEP 6 Modify your `MainActivity` code as follows:

1. Search the web for "realistic" Chicago temps for the next 5 days and add them (update) to your string array, **`wkTemps`**. Likewise include the days of the week (Monday, Tuesday, etc.) so your temperatures actually reflect temps by day, into your array.

2. Add in a title at the top of your child view stating "5 Day Chicago Forecast". Include a white font color and a light blue backcolor.

3. In your MainActivity file, replace the Toast pop up message which renders the Fahrenheit temperature currently, with a TextView which will serve to display the Fahrenheit equivalent to the *chosen* Celsius temperature by the user.

Grads

Include a "Back" Button for your second layout– with an appropriate icon for the button, that will 'simulate' the user going back to the original view to work the seek bar as depicted when the Activity first started.

Include visible views once again, to show all the elements that were included at app start up for full UX functionality, namely, the textview, seekbar, checkbox and your newly added textview added in Step 6 part 3 above which displays Fahrenheit temperature equivalencies.

So basicallly your toggling between certain element 'views' or visbilities, upon the back button's click event (Listview and back button should be not shown upon the action). Set your checkbox back to unchecked as well. Leave elements such as your textviews though in their current state (which should be still fresh with values).

STEP 7 Submitting your assignment:

For **full** credit turn the following files:

1. A pdf file of your snapshots consisting of the following:

   Your Fahrenheit result in your <u>texview</u> with a chosen temperature of 22 degrees Celsius, chosen from the seekbar.

   Your full 5 day forecast by days view that's rendered also.

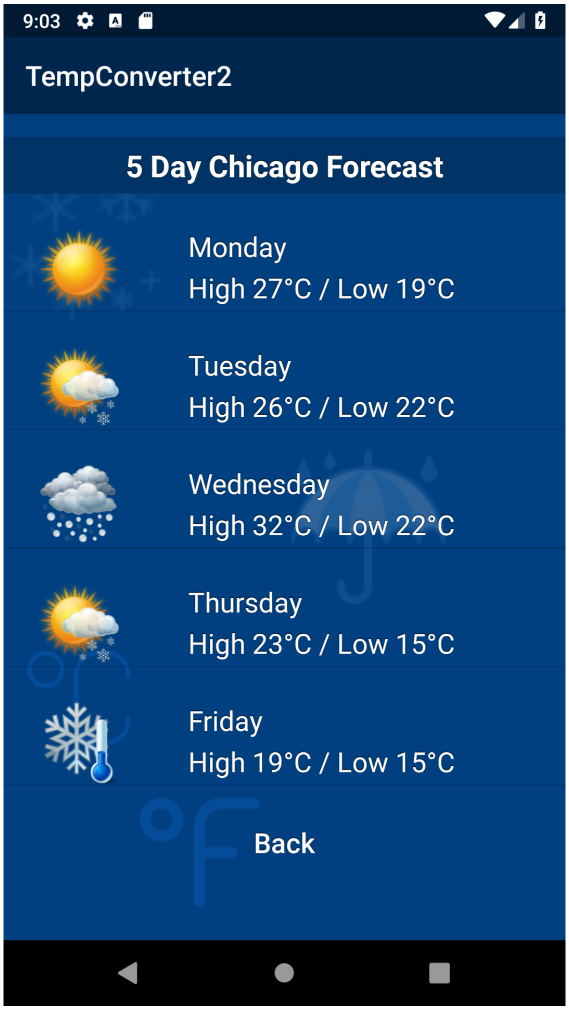
   Grads
   Include also a snapshot result of your "Back" button in action that would show the Activity screen once again with it's full UX functionality (i.e., presented in its lastest state).

2. A zip file of your MainActivity file code plus your code for <u>both</u> your XML layout files.

Extra Credit (all).

Include appropriate images for 5 day forecasts! Sample follows:

Forecast view for 5 days w. images:



5 Day Chicago Forecast

Monday
High 27°C / Low 19°C

Tuesday
High 26°C / Low 22°C

Wednesday
High 32°C / Low 22°C

Thursday
High 23°C / Low 15°C

Friday
High 19°C / Low 15°C