



S.N.J.B.'s

SHRI H. H. J. B. POLYTECHNIC, CHANDWAD

CLASS TEST I/II (201 - 201)

Course & Semester :		Roll No.:	Date : / /201
Name of Subject :		Marks obtained :	
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

Chapter 4Designing user interface with viewPackage↳ TextView → android.widget

displays text to the user & optionally allows them to edit it.

Attribute

① android:id

- uniquely identify

② android:capitalize

- 0 - don't automatically capitalize anything
- 1 - capitalize first word of each sentence
- 2 - first letter of every word
- 3 - capitalize every character

③ android:cursorVisible

Default is false (visible)

④ android:editable

true - TextView has i/p method

⑤ android:fontFamily

font family for text

⑥ android:gravity

specifies how to align text by x/y axis

(s) android:password
(t) android:phoneNumber
(u) android:textIsSelectable
(g) android:hint
hint text to display when the text is empty.

(h) android:inputType
type of data such as phone, date, time, number, passwords etc

(i) android:maxHeight / min

(j) android:maxLength / min

(k) android:text

(l) android:textAllCaps

(m) android:textColor

(n) android:textColorHighlight

(o) android:textColorHint

(p) android:textSize

(q) android:textStyle

- normal → 0
- bold → 01
- italic → 02
} separated by |

(r) android:typeface

normal → 0
sans → 1
serif → 2
monospace → 3
} separated by |

3) EditText

inherited from android.widget.TextView

Attributes

(a) android:autoText
automatically corrects some common spelling errors

(b) android:drawableBottom
to draw below the text

(c) android:drawableRight
to draw right of the text

(d) android:editable
if set, TextView

(e) android:text

inherited from android.view.View class

(a) android:background

(b) android:id

(c) android:contentDescription
briefly describes content of view

(d) android:onClick
view's context to invoke when view is clicked

(e) android:visibility
initial visibility of the view

The suggestion list is shown in dropdown menu from which a user can select in dropdown the desired item. The list of suggestions is obtained from adapter & it appears only after no. of characters that are specified in threshold.

3) AutoComplete TextView

- to show suggestions while writing in an editable Text field.

Attributes

- ① android:completionHint
- hint displayed in the dropdown menu
- ② android:completionHintView
- hint view displayed in the dropdown menu.
- ③ android:completionThreshold
- define no. of characters that user must type before completion suggestions are display in drop down menu.
- ④ android:dropdownAnchor
- view to anchor the autoComplete dropdown to
- ⑤ android:dropdownHeight
- basic height of dropdown.
- ⑥ android:dropdownHorizontalOffset
- amount of pixels by which the drop down should be offset horizontally
- ⑦ android:dropdownSelector
- selector in dropdown list
- ⑧ android:dropdownVerticalOffset
- amount of pixels by which the drop down should be offset vertically

⑨ android:dropdownWidth
- specifies the basic width of dropdown

⑩ android:popupBackground
set background

- ArrayAdapter class to display the array content

Syntax

ArrayAdapter (Context context, int resource, int textViewResourceId, T[] objects)

- context
to pass the reference of current class
- resource
to set layout (xml file) for list items in which you have text view
- textViewResourceId
to set id of TextView where you want to display the actual text
- objects
array of objects, used to set the array of elements in the text view.

eg:-

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.select_dialog_item, fruits);
```

getAdapter() returns filterable list adapter used for auto completion

setAdapter() - to set the adapter of autoCompleteTextView.

eg:- MainActivity.java

```
public class MainActivity extends Activity {
    AutoCompleteTextView o;
    String subjects = {"MAD", "MGT", "PHP", "PYN", "EDP", "ITE", "IPR"};
    onCreate()
    {
        Adapter <String> adapter = new
        ArrayAdapter <String> (this, android.R.layout.select_dialog_items, subjects);
```

```
o = (AutoCompleteTextView)findViewById(R.id.autoCompleteTextView);
```

```
o.setThreshold(1); // will start working from first character
```

```
o.setAdapter(adapter); // setting the adapter data into AutoCompleteTextView
```

```
o.setTextColor(Color.RED);
```

4) Button

push-button which can be pressed, or clicked by user to perform an action.

Attributes

① android: autoText

It specifies that this TextView has a textual i/p method & automatically corrects some common spelling errors.

② android: drawableBottom
This is drawable to be drawn below the text

③ android: drawableRight
_____ right

④ android: editable
if sets, specifies that this TextView has i/p method

⑤ android: text

⑥ android: background

⑦ android: contentDescription
- briefly describes content of view

⑧ android: id

⑨ android: onClick

⑩ android: visibility

ImageButton

This shows a button with image (instead of text) that can be pressed or clicked by the user.

① android:adjustViewBounds

set this to true if you want the Imageview to adjust its bounds to preserve the aspect ratio of its drawable

② android:baseline

offset of the baseline within this view

③ android:baselineAlignBottom

if true, image view will be baseline aligned with based on its bottom edge

④ android:cropToPadding

if true, image will be cropped to fit within its padding

⑤ android:src

sets a drawable as the content of this imageview

⑥ background, id, onClick, visibility etc

* Add Image to resources folder
 → res/drawable-dpi-folde

checkboxes

* Add ImageButton in xml file

* MainActivity

```

ImageButton i;
onCreate()
{
  addListenerOnButton();
}
public void addListenerOnButton()
{
  i = (ImageButton) findViewById(R.id.
  imagebutton);
  i.setOnClickListener(new OnClickListener()
  {
    public void onClick(View v)
    {
      Toast.makeText(this, "ImageButton is
      clicked", Toast.LENGTH_SHORT).sho
      w();
    }
  });
}
}

```

Checkbox

IS an on/off switch that can be toggled by user.

You should use checkboxes when presenting users with group of selectable option that are not mutually exclusive

android:autotext

- :drawableBottom
- :drawableRight
- :editable
- :text
- :background
- :contentDescription
- :id

Methods of CheckBox class

- ① public boolean isChecked()
 - return true if checked
- ② public void setChecked(boolean status)
 - changes state of checkbox

eg:-

```

MainActivity.java
{
  CheckBox c1, c2, c3;
  Button b1;
  onCreate()
  {
    c1 = (CheckBox) findViewById(R.id.checkbox);
    c2
    c3
  }
}

```

```

bl = (Button) findViewById(R.id.button);
bl.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v)
    {

```

```

StringBuffer sb = new StringBuffer();
sb.append("selected items");
if (c1.isChecked()) {
    sb.append("Pizza 100Rs");
    t += 100;
}
if (c2.isChecked()) {
    sb.append("Burger 120Rs");
    t += 120;
}
if (c3.isChecked()) {
    sb.append("Coffee 50Rs");
    t += 50;
}
sb.append("Total " + t);
Toast.makeText(getApplicationContext(), sb.toString(), Toast.LENGTH_LONG).show();
}
}

```

Toggle Button

displays checked/unchecked states as button. It is basically on/off button with light indicator

① android:disabledAlpha

alpha to apply to the indicator when disabled

② android:textOff

text for button when it is not checked

③ android:textOn

Methods

① CharSequence getTextOff()

return the text when button is not in checked state

② CharSequence getTextOn()

③ void setChecked(boolean checked)

changes the checked state of this button

eg:- onClick()

2 toggle button
1 button

```
StringBuffer br = new StringBuffer();
```

```
br.append(toggleButton1.getText());
```

```
Toast.makeText(getApplicationContext(), br.toString(), Toast.LENGTH_LONG).show();
```

RadioButton

is two states button which is either checked or unchecked

It is generally used with RadioGroup

up.

- RadioGroup contains several radio buttons, marking one radio button, as checked makes all other radio button as unchecked

xml.

```
<LinearLayout>
  <TextView>
  <RadioButton>
  <RadioButton2>
  </View>
  <TextView>
  <RadioGroup>
    <RadioButton1>
    <RadioButton2>
  </RadioGroup>
  <Button>
</LinearLayout>
```

MainActivity.java

```
Button b1;
RadioButton rb;
RadioGroup rg;
onCreate()
{
  rg = (RadioGroup) findViewById(R.id.
  RadioGroup);
}
```

```
public void onclickbuttonMethod (View v)
{
  int selectedId = rg.getCheckedRad
  ioButtonId()
  rb = (RadioButton) findViewById(
  selectedId);
  if (selectedId == -1)
  {
    Toast.makeText (this, "Nothing selected",
    Toast.LENGTH_SHORT).show();
  }
  else
  {
    (this, rb.getText(),
```


Progress Bar

dialog box to display the status of work being done

android.app.ProgressDialog class to show progress bar

Methods

- ① getMax() - max value of progress
- ② incrementProgressBy (int diff) - increment progress bar by diff of value
- ③ setIndeterminate (Boolean) set progress indicator as determinate or indeterminate
- ④ setMax (int max) set max value of progress
- ⑤ setProgress (int value) to update progress with specific value
- ⑥ show (Context context, CharSequence title, CharSequence Message) - display progress dialog

Range → 0 to 10000



S.N.J.B.'s
SHRI H. H. J. B. POLYTECHNIC, CHANDWAD
CLASS TEST I/II (201 - 201)

000576

Course & Semester :		Roll No. :	Date : / /201	
Name of Subject :			Marks obtained :	
Sign. of Supervisor :		Sign. of Sub. Examiner :		
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks	

```
Button b1;  
private ProgressDialog p;  
onCreate()  
{  
    b1 = (Button) findViewById(R.id.button);  
}  
public void download (View v)  
{  
    p = new ProgressDialog (this);  
    p.setCancelable (true); // you can cancel it by pressing back button  
    p.setMessage ("File Downloading");  
    p.setProgressStyle (ProgressDialog.  
        STYLE_HORIZONTAL);  
    p.setProgress (0); // initial progress is 0  
    p.setMax (100); // sets maximum value 100  
    p.show (); // display the progress bar  
  
    final int totalProgressTime = 100;  
    final Thread t = new Thread ()  
    {  
        public void run ()  
        {  
            int jumptime = 0;  
            while (jumptime < totalProgressTime)  
            {  
                try  
                {  
                    sleep (200);  
                    jumptime += 5;  
                    p.setProgress (jumptime);  
                }  
                catch (InterruptedException e) {}  
            }  
        }  
    }  
};
```

```
t.start();
```

```
}
```

```
}
```

Spinner

drop-down menu with multiple values from which the end user can select only one value

It is associated with AdapterView



Supplied By: SNJB	Course & Semester :		Roll No.:	Date : / / 201
	Name of Subject :			Marks obtained :
	Sign. of Supervisor :		Sign. of Sub. Examiner :	
	Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

List View

which contains the group of items & displays in scrollable list

It uses Adapter classes which add the content from data source (string array, array, db etc) to ListView

eg:- `<resources>`
String.xml `<string name="app-name"> ListViewDemo </string>`
`<string-array name="a_technology">`
`<item> Andeoid </item>`
`-||- Java -||-`
`-||- PHP -||-`
`</string-array>`
`</resources>`

Mainactivity

```

{
    ListView l;
    TextView v;
    String[] listitem;

    onCreate()
    {
        l = (ListView) findViewById (R.id. listView);
        v = (TextView) findViewById (R.id. TextView);
        listitem = getResources().getStringArray (R.array.
        a_technology);
        ArrayAdapter <String> adapter = new ArrayAdapter
        <String> (this, android.R.layout.simple_list_item_1
  
```

```

    listitem);
    1. setAdapter (adapter);
    1. setOnItemClickListener (new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> adapterView, View v, int position, long id) {
            String v = adapter.getItem(position);
            Toast.makeText(getApplicationContext(), v, Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

Attributes

- ① id
- ② divider: This is drawable or color to draw between list items.
- ③ dividerHeight: This specifies height of the divider
- ④ entries: Specifies the reference to an array resource that will populate the ListView
- ⑤ footerDividersEnabled: When set to false, it will not draw divider after each footer view. The default value is true
- ⑥ headerDividersEnabled: The default value is true. false → ListView will not draw divider after each header view

GridView

shows items in 2D scrolling grid (rows & column) & the grid item are not necessarily predetermined but they automatically inserted to the layout using ListAdapter.

Attributes

- ① id
- ② columnWidth: specifies the fixed width for each column.
- ③ gravity
- ④ horizontalSpacing: define the default horizontal spacing between columns.
- ⑤ numColumns: define how many columns to show
- ⑥ verticalSpacing: define the default vertical spacing between rows
- ⑦ stretchMode: define how columns should stretch to fill the available empty space
 - none
 - spacingWidth
 - columnWidth
 - spacingWidthUniform

Note - If we don't specify numColumn in GridView it behaves like ListView with SingleChoice

```
eg:- MainActivity
    § Integer list[] = {1, 2, 3, 4, ... 100};
    on(create())
```

```
    §
    GridView gv = (GridView) findViewById(R.id.
    gridView);
```

```
    ArrayAdapter<Integer> adapter = new ArrayAdapter
    <Integer>(MainActivity.this, android.R.layout.
    simp_list_item, list);
```

```
    gv.setAdapter(adapter);
```

```
    gv.setOnItemClickListener(new AdapterView
    .OnItemClickListener())
```

```
    §
    public void onItemClick(AdapterView<?> parent
    , View v, int position, long id)
```

```
    §
    Toast.makeText(MainActivity.this, "Num
    ber is:" + list[position], Toast.LENGTH_
    SHORT).show();
```

```
    }
```

```
};
```

```
}
```

```
}
```

ImageView

add image in the drawable folder

eg:-

```
public class MainActivity
{
    Button b;
    ImageView v;
    onCreate()
    {
        b = (Button) findViewById(R.id.button);
        v = (ImageView) findViewById(R.id.imageView);
        b.setOnClickListener(new OnClickListener()
        {
            public void onClick(new OnClickListener
            View v
            OnClickListener)
            {
                v.setImageResource(R.drawable
                .mango);
            }
        });
    }
}
```

Attributes

adjustViewBounds
baseline
baselineAlignBottom
cropToPadding
maxHeight
maxWidth
scr

ScrollView

scroll the child elements of palette inside scrollView

ScrollView working →

Main layout



ScrollView



again define layout tag



add widgets

Create TextView with scrolling features to insert multiple lines of text inside textview on single activity

→

```
TextView t = (TextView) findViewById(R.id
textView);
scrollable t.setMovementMethod(new
ScrollingMovementMethod());
}
```

< TextView

android:scrollbars = "vertical"

eg of scrollview

```
<LinearLayout>
  <ScrollView>
    <LinearLayout>
      <button>
      <button>
    </LinearLayout>
  </ScrollView>
</LinearLayout>
```

Custom Toast Alert

Toast is used to display information for period of time.

It contains a message to be displayed quickly & disappears after a specified period of time. It does not block the user interaction.

Toast is a subclass of Object class.

We can also create our custom toast by using a custom layout (XML file).

① makeText(Context context, CharSequence text, int duration)

It is used to initiate the toast.

LENGTH_LONG & LENGTH_SHORT are used for setting the duration for the toast.

② show()

to display the toast on screen

eg:- Toast t = Toast.makeText(getApplicationContext(), "simple toast", Toast.LENGTH_LONG);
t.show();



Course & Semester :	Roll No.:	Date : / /201	
Name of Subject :	Marks obtained :		
Sign. of Supervisor :	Sign. of Sub. Examiner :		
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

③ setGravity(int, int, int)

to set gravity for the toast
3 parameter

- 1) Gravity constant
- 2) X-position of toast
- 3) Y-position

eg:- t.setGravity(Gravity.TOP | Gravity.LEFT, 0, 0);

④ setText(CharSequence s)

to set text for toast, we want to change text value for toast then we used

eg:- t.setText("change toast text");

⑤ setDuration(int duration)

- to set duration for toast
- change duration for toast then we used

eg:- t.setDuration(Toast.LENGTH_SHORT);

⑥ inflate(int, ViewGroup)

used to inflate the layout from xml
first parameter is layout resource ID & second is root view

eg:-
LayoutInflater i = getLayoutInflater();
View layout = i.inflate(R.layout.customtoast, (ViewGroup)findViewById(R.id.toast_layout_root));

⑦ setView(View)

- to set the view for the toast
 - we pass the inflated layout which we inflate using inflate() method
- eg:- t.setView(layout);

eg: Default Toast

```
Button b = (Button) findViewById(R.id.button);  
b.setOnClickListener(new View.OnClickListener()  
{
```

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

```
    {  
        Toast t = Toast.makeText(getApplicationContext(),  
            "Default toast style", Toast.LENGTH_SHORT);  
        t.show();  
    }
```

eg:- Show Toast prompt in specified location

```
t.setGravity(Gravity.TOP, 100, 100);
```

eg:- show image in Toast prompt

```
LinearLayout l = (LinearLayout) t.getView();  
ImageView v = new ImageView(getApplicationContext().getResources().getDrawable(R.drawable.mango));  
l.addView(v, 0);  
t.show();
```

eg:- custom toast view content

```
onClick()  
{
```

```
    View toastview = getLayoutInflater().inflate(R.layout.activity_toast_custom_view, null);  
    Toast toast = new Toast(getApplicationContext());  
    toast.setView(toastview);  
    toast.setDuration(Toast.LENGTH_SHORT);  
    toast.setGravity(Gravity.CENTER, 0, 0);  
    toast.show();  
}
```


Attributes @id
 @background
 @datePickerMode → Spinner
 or calendar default
 from API 21 you have set spinner
 @padding

Date Picker

allow you to select the date consisting of day, month & year in your custom UI.

Methods

① getDayOfMonth()
 - selected day of month

② getMonth()

③ getYear()

④ setMaxDate(long maxDate)

⑤ setMinDate(long minDate)

⑥ setSpinnersShown(boolean shown)

⑦ updateDate(int year, int month, int dayOfMonth)

⑧ getCalendarView()

⑨ getFirstDayOfWeek()

eg:-

```

DatePicker picker;
Button b;
TextView t;

onCreate()
{
    t = (TextView) findViewById(R.id.textView);
    b = (Button) findViewById(R.id.button);
    picker = (DatePicker) findViewById(R.id.button);
    t.setText("Current Date" + getCurrentDate());
    b.setOnClickListener(new View.OnClickListener()
    {
        onClick(View v)
        {
            t.setText("change date" + getCurrentDate());
        }
    });
}

public String getCurrentDate()
{
    StringBuilder/ StringBuffer br = new StringBuffer();
    br.append(picker.getMonth() + 1 + "/");
    br.append(picker.getDayOfMonth() + "/");
    br.append(picker.getYear());
    return br.toString();
}
}

```

TimePicker

to select date
It allows you to select time by hour & minute.
You can't select time by second

Methods

- ① is24HourView()
true if this is in 24hr view
- ② isEnabled()
return enabled status for this view
- ③ setCurrentHour(Integer currentHour)
set the current hour
- ④ setCurrentMinute(Integer currentMinute)
set current minute
- ⑤ setEnabled(Boolean enabled)
set enabled state of this view
- ⑥ setIs24HourView(Boolean is24HourView)
set whether in 24 Hr or AM/PM mode
- ⑦ ~~setOnTimeCh~~
setOnTimeChangeListener(TimePicker.OnTimeCh
echangedListener onTimeChangeListener)
set the callback that indicates the
time has been adjusted by the user
- ⑧ getCurrentHour()
to get the time selected by the user
(hr)
- ⑨ getCurrentMinute()
to get the time selected by the user
(minute)

Attribute

- ① id
- ② timePickerMode → clock/spinner
default → after 24hr set
- ③ background
- ④ padding

eg:-

```
MainActivity
```

```
{  
    TextView t;  
    Button b;  
    int hr, min;  
    onCreate()  
    {  
        t=(TextView)findViewById(R.id.TextView);  
        b=(Button)findViewById(R.id.button);  
        Calendar c=Calendar.getInstance();  
        hr=c.get(Calendar.HOUR_OF_DAY);  
        min=c.get(Calendar.MINUTE);  
        updateTime(hr, min);  
        b.setOnClickListener(new OnClickListener()  
        {  
            public void onClick(View v)  
            {  
                showDialog().show();  
            }  
        });  
    }  
    protected Dialog showDialog(int id)  
    {  
        switch(id)  
        {  
            case TIME_DIALOG_ID:  
                return new TimePickerDialog(this, timePic  
kerListener, hr, min, false,  
                );  
            return null;  
        }  
    }  
    private TimePickerDialog.OnTimeSetListener tp=new  
    TimePickerDialog.OnTimeSetListener()  
    {  
        }  
    }  
}
```

```

public void onTimeSet(TimePicker view, int hourOfDay,
    int minutes)
{
    hr = hourOfDay;
    min = minutes;
    updateTime(hr, min);
}
}

private void updateTime(int hours, int mins)
{
    String timeset = "";
    if (hours > 12)
    {
        hours -= 12;
        timeset = "pm";
    } else if (hours == 0)
    {
        hours += 12;
        timeset = "Am";
    } else if (hours == 12)
    {
        timeset = "pm";
    }
    else
    {
        timeset = "Am";
    }
    String minutes = "";
    if (mins < 10)
        minutes = "0" + mins;
    else
        minutes = String.valueOf(mins);
    String s = new StringBuilder().append
    (hour).append(":").append(minutes).append
    (" ").append(timeset).toString();
    t.setText(s);
}

private static String utilTime(int value)
{
    if (value < 10) return "0" + String.valueOf(value);
    else return String.valueOf(value);
}
}

```

Exp No - 16. Date & Time Picker

```

1) ① android:timePickerMode="clock">
    ② android:timePickerMode="spinner"> 24 hours
    we can change TimePicker in spinner mode to
    AM/PM format
    TimePicker t = (TimePicker) findViewById(R.id.
    timepicker);
    t.setIs24HourView(true);

    onClick()
    {
        int h, m;
        String am_pm;
        if (Build.VERSION.SDK_INT >= 23)
        {
            h = t.getHours();
            m = t.getMinutes();
        }
        else
        {
            h = t.getCurrentHour();
            m = t.getCurrentMinute();
        }
        if (h > 12)
        {
            am_pm = "PM";
            h = h - 1;
        }
        else
        {
            am_pm = "AM";
        }
        tv.setText("Selected Date" + h + ":" + m +
        am_pm);
    }
}

```

```

② Button bd, bt;
   EditText td, tt;
   private int y, m, d, h, m;
   public void onclick(View v)
   {
       if (v == bd)
       {
           final Calendar c = Calendar.getInstance();
           y = c.get(Calendar.YEAR);
           m = c.get(Calendar.MONTH);
           d = c.get(Calendar.DAY_OF_MONTH);
           DatePickerDialog dpg = new DatePickerDialog(
               this, new DatePickerDialog.OnDateSetListener()
           {
               public void onDateSet(DatePicker view,
                   int year, int monthOfYear, int dayOfMonth)
               {
                   td.setText(dayOfMonth + "-" + (month
                       ofYear + 1) + "-" + year);
               }
           }, y, m, d);
           dpg.show();
           if (v == bt)
           {
               final Calendar c = Calendar.getInstance();
               h = c.get(Calendar.HOUR_OF_DAY);
               m = c.get(Calendar.MINUTE);
               TimePickerDialog tpg = new TimePickerDialog(
                   this, new TimePickerDialog.OnTimeSetList
                       ener()
               {
                   public void onTimeSet(int TimePicke
                       view, int hourOfDay, int minute)
                   {
                       tt.setText(hourOfDay + ":" + minute);
                   }
               }, h, m, false);
               tpg.show();
           }
       }
   }

```