Chapter 3:

·       14, creates a GridLayout within the current context and sets its number of rows to four and its number of columns to two.

GridLayout gd =  new GridLayout(this);

gd.setColumnCount(2);

gd.setRowCount(4);

·       15, This code create a button within the current context.

Button button = new Button(getContext());

·       16, This code creates a 5 \* 2 two-dimensional array of buttons within the current context.

Button [][]  buttons = new Button[5][2];

for( int i = 0; i < 5;  i++) {

         for(int  j=0; j < 2 ; j ++) {

                     buttons[i][j] = new Button(getContext());

         }

}

·       17, this code adds a Button object named b, specifying its width and heights as 200 pixels each, to an already created GridLayout object named gl.

Button b = new Button(getContext);

b.setLayoutParams(new LinearLayout.LayoutParams(200, 200));

gl.addView(b);

·       21, This code checks if the button that was clicked is a button name b. If it is, it outputs to Logcat YES, otherwise, it outputs to Logcat NO.

Private class ButtonHandler implements View.OnClickListener {

         Private static final String TAG = “LOG”;

         Public void onClick(View v) {

                     If(v == b) {

                                 Log.i(TAG , “YES”);

                     }else{

                                 Log.i(TAG  G, “NO”);

         }

}

Chapter 4:

·       1, The TableLayout class can be used to organize various GUI conpoents: **As a table of rows and columns.**

·       2, The direct superclass fo LinearLayout and RelativeLayout is **ViewGroup**

·       3, TableLayout and TalbeRow are direct subclasses of **LinearLayout**

·       4, The RelativeLayout class is a good choice to organize various GUI components **So that we position components relative to other components**

·       5, in **android.content** package is the intent class

·       6, After you have created an intent for a new activity, **startActivity**method of the Activity class you call with that intent parameter in order to start a new activity?

·       7, **onRestart** method of the activity class is automatically called when an activity is about to restart?

·       8, What methods of the Activity class (and in what order) are automatically called when an activity is first created?

**On create, on start and onResume**

·       9, **onStop** method of the activity class is automatically called when an activity becomes invisible to the user?

·       10, Two activities can share the same data?

o  **Yes, for example by each accessing a public static instance variable from another class.**

Chapter 7

1 A.setOnTouchListener

2.B getAction

3. C bringToFront

4. B GestureDetector

5. C Public static inner interfaces of GestureDetector

6. D Constants that the action can be compared to

7. B onTouchEvent

18

protected void onCreate( Bundle savedInstanceState ){

   super.onCreate( savedInstanceState );

   // Your code goes here

   GestureAndTapHandler gth = new GestureAndTapHandler( );

   detector = new GestureDetector( this, gth );

   detector.setOnDoubleTapListener( gth );

    detector.onTouchEvent( event );

}

19

public boolean onTouchEvent( MotionEvent event ){

  // Your code goes here

   detector.onTouchEvent(event)

}