Chapter 3

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

```
Context context = getApplicationContext();
GridLayout gridLayout = new GridLayout(context);
gridLayout.setRowCount(4);
gridLayout.setColumnCount(2);
```

15. This code creates a button within the current context.

```
Context context = getApplicationContext();
Button button = new Button(context);
```

16. This code creates a 5 × 2 two-dimensional array of buttons within the current context.
public class MainActivity extends AppCompatActivity {

```
    Context context = getApplicationContext();

    Button[][] buttons = new Button[5][2];
    for (int i = 0; i < buttons.length; i++) {
       for (int j = 0; j < buttons[0].length; j++) {
          buttons[i][j] = new Button(context);
       }
    }
```

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

```
Context context = getApplicationContext();
Button b = new Button(context);
b.setWidth(200);
b.setHeight(200);
gl.addView(button);
```

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

```
button.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View view) {
        If (view.getText() == b) {
        Log.i("YES");
  }
        else {
        Log.i("NO");
});
```

Chapter 4

1. The TableLayout class can be used to organize various GUI components
As a table of rows and columns

As a table of rows and columns
As a table of multiple rows with only one column each
As a table of only one row and multiple columns
As a table of only one row and one column

2. The direct superclass of LinearLayout and RelativeLayout is **ViewGroup**
View
ViewGroup
Layout
Object

3. TableLayout and TableRow are direct subclasses of
LinearLayout

LinearLayout
ViewGroup
RelativeLayout
View

4. The RelativeLayout class is a good choice to organize various GUI components

So that we position components relative to other components

To give the components absolute x- and y-coordinates
So that we position components relative to other components
As a grid of multiple rows and columns
It is never a good choice

5. In what package is the Intent class?
android.content

java.intent
android.widget
android.activity
android.content

6. After you have created an Intent for a new activity, what method of the Activity class do you call with that Intent parameter in order to start a new activity?
startActivity

startActivity
newActivity
startIntent
newIntent

7. What method of the Activity class is automatically called when an activity is about to restart?
onRestart

onCreate
onDestroy
onRestart
onGo

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

onCreate
onCreate, onStart, and onResume (in that order)
onCreate and onResume
onStart, onCreate, and onResume (in that order)

9. What method of the Activity class is automatically called when an activity becomes invisible to the user?
onStop

onResume
onStop
onPause
onInvisible

10. Two activities can share the same data
Yes, for example by each accessing a public static instance variable from another class

No, it is not possible
Yes, but it is only possible by writing to and reading from the same file
Yes, but it is only possible by writing to and reading from a SQLite database
Yes, for example by each accessing a public static instance variable from another class

Chapter 7
1. What method do we use to register a View.OnTouchListener on a component?
setOnTouchListener

setOnTouchListener
addOnTouchListener
registerOnTouchListener
isOnTouchListener

2. What method of the MotionEvent class do we use to retrieve the type of action that just Happened?
getAction

action
getAction
getEvent
getTouch

3. What method do we use to bring a View to the top of the stacking order?
bringToFront

bringToTop
gotToTop
bringToFront
bringChildToTop

4. What class can be used to capture gestures and tap events?
GestureDetector

Gesture
GestureDetector
TapDetector
GestureAndTapDetector

5. OnGestureListener and OnDoubleTapListener are
Public static inner interfaces of GestureDetector

Private static inner classes of GestureDetector
Private static inner interfaces of GestureDetector
Public static inner interfaces of GestureDetector
Classes independent of GestureDetector

6. In order to identify a touch event action, the MotionEvent class has
Constants that the action can be compared to

A special constructor
Private instance variables
Private methods
Constants that the action can be compared to

7. What method of the GestureDetector class acts as a dispatcher to the various methods of OnGestureListener and OnDoubleTapListener?
onTouchEvent

onTouch
onTouchEvent
onMotionTouchEvent
onEvent

18. We are coding inside the onCreate method of an Activity class. Write the code so that the current Activity will handle the gestures and tap events.

```
Protected void onCreate (Bundle savedInstanceState ) {
        super.onCreate ( savedInstanceState);
        d = new GestureDetector(getApplicationContext(), this);
        d.setOnDoubleTapListener(this);
}
```

19. We are coding inside the onTouchEvent method of an Activity class. Write the code so that if there is a gesture event, it gets dispatched to the appropriate method of GestureDetector.OnGestureListener.

```
public boolean onTouchEvent (MotionEvent event) {
        d.onTouchEvent(event);
        return true;
}
```