

# FIT2081: Mobile application development

Semester 1, 2019

Revision Notes

# Table of Contents

<b>Mobile Applications</b>	3
Definitions	3
<b>Android</b>	4
Fragmentation	4
Android Components	4
<b>Activities</b>	5
Back Stacks	5
Activity Lifecycle	5
Activity Instance States	6
Saving/Restoring Instance State	6
<b>Switching Between Activities</b>	7
Intent	7
The Intent Object	7
Bundles	7
Adding values into the Intent	7
Getting values from the Intent	7
<b>UIs, Layouts &amp; Views</b>	7
Types of Views	8
Views	8
ViewGroup	8
Creating UIs	8
Layouts	8
Constraint Layouts	9
Adjusting View Size	9
<b>RecyclerView</b>	10
Adding RecyclerView to Layout	10
Adding an Adapter	11
<b>Content Providers</b>	11
Adding CPs to Application	12
<b>Content Resolvers</b>	13
<b>CursorLoader</b>	14
Using CursorLoaders	14
onCreateLoader	14
onLoadFinished	14
onLoadReset	14
<b>Fragments</b>	15
Differences between Fragments and Activities	15
Benefits of Fragments	15

Fragment Lifecycle	16
Example: Creating a Fragment and Adding it to An Activity	17
<b>Gestures and Motion</b>	18
Common Gestures	19
Gathering Data	19
Capturing Events to a Single View	20
MotionEvent	20
Gestures	21
GestureDetector	21
Capturing Gestures	21
onScroll	21
onFling	21
ScaleGestureDetector	22
onTouchEvent	22
<b>AsyncTask</b>	23
doInBackground()	23
onPostExecute	24
Execute()	24
<b>WebViews</b>	25
Adding WebView to Activity	25
Using the WebView Client	25
<b>Mock Exam</b>	26
<b>Workshop Quizzes from some random thingy</b>	28
<b>Workshop Quizzes (2019)</b>	30
Week 2	30
Week 3	33
Week 4	36
Week 5	38
Week 6	40
Week 7	42
Week 8	43
Week 9	45
Week 10	47
Week 11	48
Week 12	50

# Mobile Applications

## Definitions

- Software Development Kit (SDK): bundle of all software components necessary to develop and deploy on a certain platform (Java Software Development Kit - JDK , Android SDK)
- Class Library or *Application Programming Interface (API)*
- IDE
- There are 3 different types of mobile apps:
  - Native
  - Mobile Web
  - Hybrid

# Android

## Fragmentation

- Developers must deal with Fragmentation
- **Forward Compatibility:** old apps running on new platform versions, essentially considering
- **Backwards Compatibility:** newer apps running on older platform versions. (Apps are not necessarily backwards compatible, especially when using newer functionality only available on newer platforms.)

## Android Components

Component	Description
Activities	An activity is a single, standalone module of application functionality which is usually the UI screens and the screens functionality
Services	Are process that run in the background that does not have an UI
Content Providers	implements a mechanism for sharing of data between apps, manages a shared set of app data
Content resolver	Is single global instance in your application that provides access to yours and other apps content providers
Broadcast Receivers	Are mechanism by which apps are able to respond to Broadcast intents
Intents	

# Activities

Activities have 4 possible Lifecycle states:

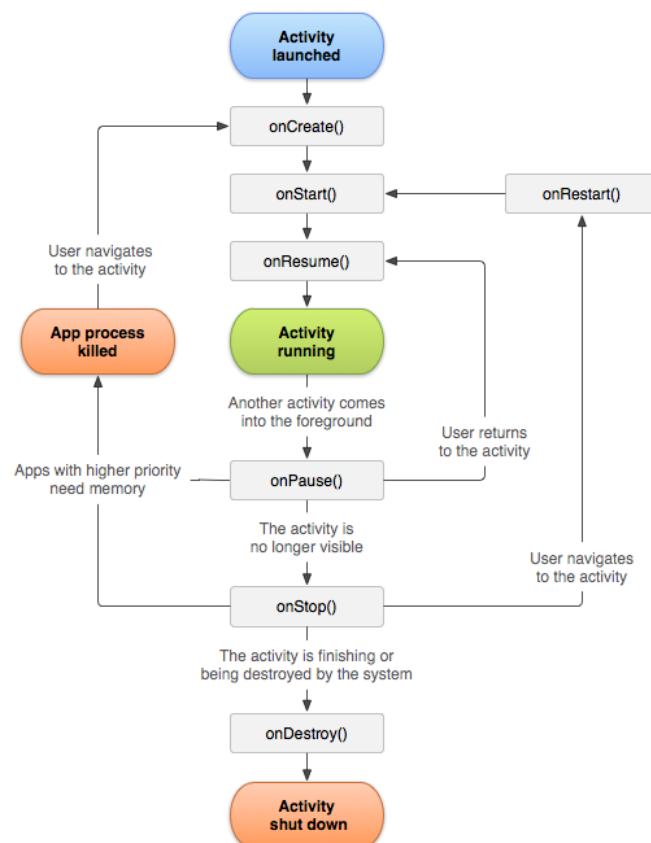
- Foreground
- Partially hidden
- Fully Hidden
- Destroyed

## Back Stacks

- The back stack is a stack of activities that the user interacts with.
- Everytime the user navigates to a new activity, the current activity gets pushed onto the stack (and retains the state)
- When the user clicks the back button, the current activity is popped from the stack

## Activity Lifecycle

- Activity States
- Lifecycle Callbacks (onCreate(), onStart(), onResume(), onPause(), onStop(), onDestroy())



- Always use `super()` in all lifecycle callbacks (this ensures that the superclass method is called)

## Activity Instance States

Activity Data:

- Instance Data (data associated with single of Activity)
- Persistent data (data associated with multiple uses of Activities)

### Saving/Restoring Instance State

Function	Notes
<code>onSaveInstanceState(Bundle outState)</code>	<ul style="list-style-type: none"><li>• Called if user indicates they are done with Activity</li><li>• Must call <code>super</code> to let Android save view-state (optional)</li><li>• TODO: write code to save non-view state if required</li></ul>
<code>onCreate(Bundle savedInstanceState)</code>	<ul style="list-style-type: none"><li>• Must call <code>super</code> to let Android restore previously saved view-state (required)</li><li>• TODO: write code to restore non-view state from Bundle (check bundle is non-null)</li></ul>
<code>onRestoreInstanceState(Bundle savedInstanceState)</code>	<ul style="list-style-type: none"><li>• Must call <code>super</code> to let Android restore previously saved view-state (required)</li><li>• TODO: write code to save non-view state from Bundle (check bundle is non-null)</li></ul>

# Switching Between Activities

## Intent

You can start a new activity with an Intent object, then calling

```
public void startActivity(View view){
    Intent i = new Intent(this, MyNewActivity.class);
    EditText editText = (EditText) findViewById(R.id.editText);
    i.putExtra("KEY", editText.getText().toString());
    startActivity(i);
}
```

## The Intent Object

Intent's constructor typically follows: `Intent(Context context, Class<?> class)` where context in an Activity can be called using the `this` object, otherwise you can also use `getContext()`

## Bundles

You can pass data by passing bundles into an Intent object which allows you to pass data between Activities. Bundles are JSON/Dictionary objects with a key/value pair.

## Adding values into the Intent

In order to add values to the intent call, you just add extra things to the Intent object.

```
intent.putExtra("key", "value");
```

## Getting values from the Intent

Then from the new Activity (`MyNewActivity.class`), you can get the values directly using `getStringExtra`, `getIntExtra`, etc.

```
@Override
public View onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    startActivity(R.layout.MyNewActivity);

    Intent intent = getIntent();
    String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
}
```

# UIs, Layouts & Views

Android UIs are made up of hierarchy of view objects



# Types of Views

## Views

Class represents the basic building block for UI components (is the base package used for widgets - *interactive UI components*)

## ViewGroup

- Base class for layouts
- Allow Views to be nested as a hierarchy (or tree, like html/xml)

### **ViewGroup.LayoutParams**

- Tell parents how they want to be laid out.
  - FILL\_PARENT (named MATCH\_PARENT in API level >8)
  - WRAP\_CONTENT (just big enough to enclose its content)
  - An exact number

## Creating UIs

You can use **Java** or **XML** in creating UIs

## Layouts




- ViewGroup
  - Has several direct subclasses (i.e. CoordinatorLayout, FrameLayout, LinearLayout, RelativeLayout)
  - Has several indirect subclasses
- View Containers
  - Many direct & indirect View Container subclasses that can be part of UI's View Hierarchy (i.e. Toolbar)
- ConstraintLayout

## Constraint Layouts

- Basically like RelativeLayout but...
  - Views are attached to layout sides or horizontal and vertical guidelines (like virtual layout sides)
  - Springs can expand and collapse depending of viewport of device they are displayed on
  - Tension of springs holding a View between 2 endpoints can be biased towards one end by percentage
  - Hard margins can be specified at end point of each spring

### Adjusting View Size

Adjustments to how Width and Height are calculated by clicking symbols:

Symbol	Attribute	Description
	Fixed	Specific dimension in layout_width and layout_height fields
	Wrap Content	View expands only as much as needed to fit its content
	Match Constraints	View expands as much as possible to meet the constraints on each side (after accounting for the view's margins)

### Constraint Bias

Constraint Bias can be adjusted by the scroll sliders along the vertical and horizontal.

# RecyclerView

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

## Introduction

- Display a scrolling list of elements based on large data sets

## Adding RecyclerView to Layout

- Add the resource to the XML layout

```
<?xml version="1.0" encoding="utf-8"?>
<!-- A RecyclerView with some commonly used attributes -->
<android.support.v7.widget.RecyclerView
    android:id="@+id/my_recycler_view"
    android:scrollbars="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

- Setup the layout as part the onCreate function of the layout

```
@Override
public View onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.MyActivity);

    // Associate reference to recycler view
    recyclerView = (RecyclerView) findViewById(R.id.my_recycler_view);

    // use this setting to improve performance if you know that changes
    // in content do not change the layout size of the RecyclerView
    recyclerView.setHasFixedSize(true);

    // use a linear layout manager
    layoutManager = new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);

    // specify an adapter (see also next example)
    mAdapter = new MyAdapter(myDataset);
    recyclerView.setAdapter(mAdapter);
}
```

-

## Adding an Adapter

The adapter communicates with the RecyclerView and Layout Manager and creates views for objects

```
public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {
    private String[] mDataset;
    // Provide a reference to the views for each data item
    public static class MyViewHolder extends RecyclerView.ViewHolder {
        // each data item is just a string in this case
        public TextView textView;
        public MyViewHolder(TextView v) {
            super(v);
            textView = v;
        }
    }

    // Provide a suitable constructor (depends on the kind of dataset)
    public MyAdapter(String[] myDataset) {
        mDataset = myDataset;
    }

    // Create new views (invoked by the layout manager)
    @Override
    public MyAdapter.MyViewHolder onCreateViewHolder(ViewGroup parent,
                                                    int viewType) {
        // create a new view
        TextView v = (TextView) LayoutInflater.from(parent.getContext())
            .inflate(R.layout.my_text_view, parent, false);
        MyViewHolder vh = new MyViewHolder(v);
        return vh;
    }

    // Replace the contents of a view (invoked by the layout manager)
    @Override
    public void onBindViewHolder(MyViewHolder holder, int position) {
        // - get element from your dataset at this position
        // - replace the contents of the view with that element
        holder.textView.setText(mDataset[position]);
    }

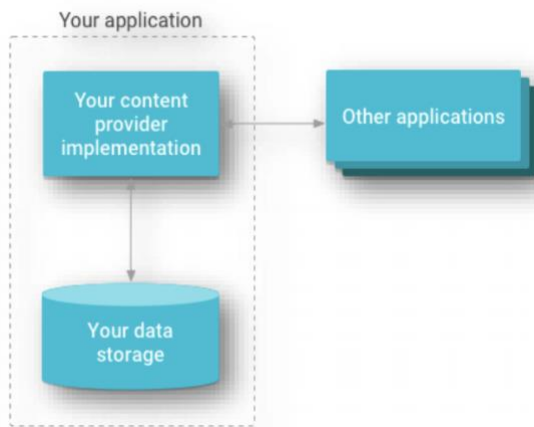
    // Return the size of your dataset (invoked by the layout manager)
    @Override
    public int getItemCount() {
        return mDataset.length;
    }
}
```

## Content Providers

- Implement a mechanism for sharing of data between apps

- Help an application manage access to data stored by itself, stored by other apps and provide a way to share data with other apps
- Can provide other applications with access

How Content Providers Manage Access to Storage:



## Adding CPs to Application

1. Add `<provider>` component to app manifest
2. Create `ContentProvider`
  - a. Create new class which extends `ContentProvider`
  - b. Create `UriMatcher` definition (Utility class to add in matching URIs in content providers)
3. Implement other functions

Lifecycle Function	What the function does
<code>onCreate()</code>	Called during initialisation the provider
<code>query(Uri, String[], Bundle, CancellationSignal)</code>	Returns data to the caller
<code>insert(Uri, ContentValues)</code>	Inserts new data into content provider
<code>update(Uri, ContentValues, String, String[])</code>	Updates existing data into content provider
<code>delete(Uri, String, String[])</code>	Delete data from content provider
<code>getType(Uri)</code>	Returns MIME type of data in CP

# Content Resolvers

- CR object is used to access the content provider
- Application can get reference by making a call to `getContentResolver()` method of context
- Content Resolver object contains set of methods that mirror Content Provider
- application simply makes calls to the methods, specifying the URI of the content on which the operation is to be performed.
- content resolver and content provider objects then communicate to perform the requested task on behalf of the application

```
// From a fragment
ContentResolver resolver =
getActivity().getApplicationContext().getContentResolver()

// from an activity
ContentResolver resolver = getContentResolver();

// resolvers can perform Content Provider methods like insert
resolver.insert(SchemeSchapes.Shape.CONTENT_URI, newContentValues);
```

# CursorLoader

- A loader that queries the ContentResolver and returns a Cursor (pointer)
- Only query (select) your data. No insert, no delete, no update
- A CursorLoader must be built with full information for query to perform

## Using CursorLoaders

1. In fragment, implement LoaderCallback interface

```
public class ArticlesFragment extends Fragment implements  
LoaderManager.LoaderCallbacks<Cursor>
```

2. Implement its 3 methods:

- a. onCreateLoader(int id, Bundle bundle)
- b. onLoadFinished(Loader loader, Cursor cursor)
- c. onLoaderReset(Loader loader)

### onCreateLoader

- Instantiate and return a new Loader for the given ID

### onLoadFinished

- Method is called when previously created loader has finished its load
- Method is typically client will update apps UI with loaded data

### onLoadReset

- method is called when a previously created loader is being reset, thus making its data unavailable.
- onLoadReset is called when the Loader's data is about to be reset. This method gives you the opportunity to remove any references to old data that may no longer be available.

# Fragments

- Reusable layouts that can be used across multiple Activities
- Fragment represents a behaviour or a portion of user interface in a `FragmentActivity`

## Differences between Fragments and Activities

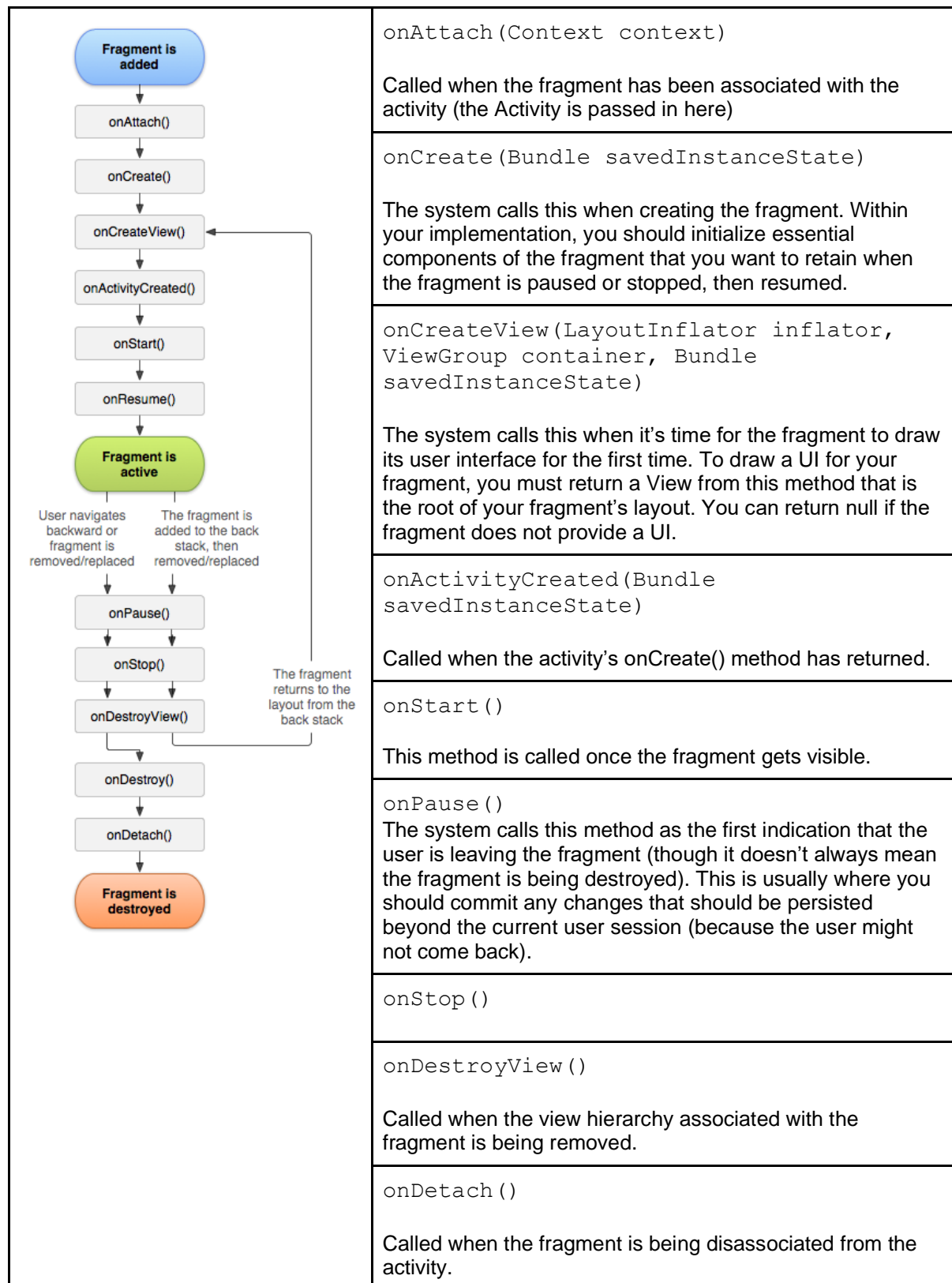
- Activity is an application component that represents a full screen. A fragment is a portion of user interface in an activity.
- An activity may contain 0 or multiple fragments.
- Fragments can be reused in multiple activities.
- A fragment can't exist independently. It should be always part of an activity.
- A fragment can be added or removed while the activity (the host) is running.
- A fragment has its own lifecycle events.

## Benefits of Fragments

- UI organisation (especially with tabs and dropdown lists)
- Passing data between screens (fragments)
- Backstack manipulation (Using `FragmentManager`, its easy to clear all Fragments, than insert)
- Reusability



# Fragment Lifecycle



## Example: Creating a Fragment and Adding it to An Activity

### 1. Create two fragments

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.ViewGroup;

public class ArticlesFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState){
        // Inflate layout
        return inflater.inflate(R.id.article_view, false);
    }
}
```

```
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.ViewGroup;

public class HeadlinesFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState){
        // Inflate layout
        return inflater.inflate(R.id.headlines_view, false);
    }
}
```

### 2. Add a container (FrameLayout) to activity's layout

### 3. Add Fragment to created Layout

```
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState){
        // Call original function
        super.onCreate(savedInstanceState);
        setContentView(R.id.news_articles);

        // create new Fragment
        HeadlinesFragment headlinesFragment = new
HeadlinesFragment();

        // add fragment to layout

        getFragmentManager().beginTransaction().add(R.id.fragment_container
, headlinesFragment).commit();
    }
}
```

You can also pass data from activity/fragment using Bundle by setting it as arguments using `fragment.setArguments(Bundle bundle)`

# Gestures and Motion

Note: pointers refers to fingers

MotionEvent	Description
ACTION_DOWN	For first pointer (fingers) touches screen, index always at 0
ACTION_POINTER_DOWN	Extra pointers enters screen beyond the first, the pointer data at index, pointer data returned by <code>getActionIndex()</code>
ACTION_MOVE	Change has happened during press gesture
ACTION_POINTER_UP	Sent when non-primary pointer leaves screen
ACTION_UP	Sent when last pointer leaves screen

## Common Gestures

Touch gesture occurs when a user places one or more fingers on the touch screen and app.

1. [Gather Data about touch events](#)
2. Interpret data

### Gathering Data

When a user places 1 or more fingers on the screen, this triggers the callback of `onTouchEvent(MotionEvent event)` on the View. For sequence of touch events, this is ultimately identified as a gesture, `onTouchEvent()` is fired several times.

```
public class MainActivity extends Activity {
    ...
    // This example shows an Activity, but you would use the same
    approach if
    // you were subclassing a View.
    @Override
    public boolean onTouchEvent(MotionEvent event){

        int action = MotionEventCompat.getActionMasked(event);

        switch(action) {
            case (MotionEvent.ACTION_DOWN) :
                Log.d(DEBUG_TAG, "Action was DOWN");
                return true;
            case (MotionEvent.ACTION_MOVE) :
                Log.d(DEBUG_TAG, "Action was MOVE");
                return true;
            case (MotionEvent.ACTION_UP) :
                Log.d(DEBUG_TAG, "Action was UP");
                return true;
            case (MotionEvent.ACTION_CANCEL) :
                Log.d(DEBUG_TAG, "Action was CANCEL");
                return true;
            case (MotionEvent.ACTION_OUTSIDE) :
                Log.d(DEBUG_TAG, "Movement occurred outside bounds " +
                    "of current screen element");
                return true;
            default :
                return super.onTouchEvent(event);
        }
    }
}
```

## Capturing Events to a Single View

You can also set a `View.OnTouchListener` object to any `View` object using the `setOnTouchListener()` method.

```
View myView = findViewById(R.id.my_view);
myView.setOnTouchListener(new OnTouchListener() {
    public boolean onTouch(View v, MotionEvent event) {
        // ... Respond to touch events
        return true;
    }
});
```

If a listener returns false for the `ACTION_DOWN` event

## MotionEvent

Motion events describe movements in terms of an action code and set of axis values.

- Action code specifies the state change (pointer going up or down)
- Axis values describe position and other movement properties

# Gestures

## GestureDetector

The GestureDetector can be used on Activities by creating a Listener which extends off GestureDetector.SimpleOnGestureListener.

### Capturing Gestures

The listener can capture different types of gestures

#### onScroll

Dragging is the type of scrolling that occurs when a user drags her finger across the touch screen (onScroll).

```
public abstract boolean onScroll (  
    MotionEvent e1,  
    MotionEvent e2,  
    float distanceX,  
    float distanceY)
```

#### onFling

onFling is the type of scrolling that occurs when a user drags and lifts her finger quickly.

```
public abstract boolean onFling (  
    MotionEvent e1, # first motion  
    MotionEvent e2, # current triggered motion  
    float velocityX, #velocity measured in px along x-axis  
    float velocityY) #velocity measured in px along y-axis
```

## ScaleGestureDetector

ScaleGestureDetector detects scaling transformation gestures using the supplied MotionEvent. The OnScaleGestureListener callback will notify users when a particular gesture event has occurred. This class should only be used with MotionEvent reported via touch. To use this class, you will need to do the following

- Create an instance of the ScaleGestureDetector for your [View](#)
- In the View#onTouchEvent(MotionEvent) method ensure you call onTouchEvent(android.view.MotionEvent). The methods defined in your callback will be executed when the events occur.

## onTouchEvent

onTouchEvent is a way to detect touch gestures and detect dragging an object across the screen. Applications should pass a complete and consistent event stream to this method. A complete and consistent event stream involves all MotionEvent from the initial ACTION\_DOWN to the final ACTION\_UP or ACTION\_CANCEL.

# AsyncTask

Asynchronous Tasks help the application continue running whilst your code is doing something in the background such as fetching data.

Typically AsyncTask you will need to implement (Override) the following operations: `doInBackground()`, `preExecute()`, `postExecute()`, `onProgressUpdate()`.

## doInBackground()

```
protected Class<?> doInBackground(String ...params)
```

This function tells what the AsyncTask to do in the background, it can take many parameters (as String). In the example below (Week 10 Lab/Tute) we do an HTTP API Request to get information about a country, we then return that as part of the model.

```
@Override  
protected CountryInfo doInBackground(String... params) {  
    CountryInfo countryInfo = null;  
    try {  
        // Create URL  
        String selectedCountry = params[0];  
        URL webServiceEndPoint = new URL("https://restcountries.eu/rest/v2/name/" +  
selectedCountry); //  
  
        // Create connection  
        HttpsURLConnection myConnection = (HttpsURLConnection)  
webServiceEndPoint.openConnection();  
  
        if (myConnection.getResponseCode() == 200) {  
            //JSON data has arrived successfully, now we need to open a stream to it and get a  
reader  
            InputStream responseBody = myConnection.getInputStream();  
            InputStreamReader responseBodyReader = new InputStreamReader(responseBody,  
"UTF-8");  
  
            //now use a JSON parser to decode data  
            JsonReader jsonReader = new JsonReader(responseBodyReader);  
            jsonReader.beginArray(); //consume arrays's opening JSON brace  
            String keyName;  
            countryInfo = new CountryInfo(); //nested class (see below) to carry Country Data  
around in  
            boolean countryFound = false;  
            //process array of objects  
            while (jsonReader.hasNext() && !countryFound) {  
                jsonReader.beginObject(); //consume object's opening JSON brace  
                while (jsonReader.hasNext()) { // process key/value pairs inside the current object  
                    keyName = jsonReader.nextName();  
                    // handle JSON  
                }  
                jsonReader.endObject();  
            }  
            jsonReader.endArray();
```



```

        } else {
            Log.i("INFO", "Error: No response");
        }

        // All your networking logic should be here
    } catch (Exception e) {
        Log.i("INFO", "Error " + e.toString());
    }
    return countryInfo;
}

```

## onPostExecute

protected void onPostExecute(Object<?> object)

The onPostExecute function tells the AsyncTask how to handle the object, the additional parameter has the type that you returned in the doInBackground() method.

```

@Override
protected void onPostExecute(CountryInfo countryInfo) {
    super.onPostExecute(countryInfo);
    name.setText(countryInfo.getName());
    capital.setText(countryInfo.getCapital());
    code.setText(countryInfo.getAlpha3Code());
    population.setText(Integer.toString(countryInfo.getPopulation()));
    area.setText(Double.toString(countryInfo.getArea()));
}

```

## Execute()

AsyncTask.execute(String ..params)

The execute method starts the asynchronous task.

# WebViews

Webviews are views that display websites. You can also use the `WebViewClient` class to help override the action where the user loads the web browser when they click a link

## Adding WebView to Activity

You can add websites directly to the layout of the Activity you want to run

```
<WebView  
    android:id="@+id/webview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

## Using the WebView Client

You can set the `WebView` client to prevent the user loading a web browser when it detects a link click.

```
WebView myWebView = (WebView) findViewById(R.id.webview);  
myWebView.setWebViewClient(MyWebViewClient);
```

Then to load a website use the `WebView.loadUrl(String url)` method

```
WebView myWebView = (WebView) findViewById(R.id.webview);  
myWebView.loadUrl("http://www.example.com");
```

# Mock Exam

1. With respect to providing a platform on which to execute a mobile app, what does a thin native client and a mobile Web browser have in common?

*Thin native clients*

2. List two scenarios where Android will not save the view-state of an activity.
3. What is the datatype of MainActivity.EXTRA\_MSG? What is its purpose?

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

    Intent intent = getIntent();
    String msg =
intent.getStringExtra(MainActivity.EXTRA_MSG);
    TextView tV = findViewById(R.id.textView);
    tV.setText(msg);
}
```

*The data type of MainActivity.EXTRA\_MSG is string, the main purpose of this is the key of the key/value pair in the intent provided by the activity*

4. What data format does the Intent class use to store data? Briefly explain.

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

    Intent intent = getIntent();
    String msg =
intent.getStringExtra(MainActivity.EXTRA_MSG);
    TextView tV = findViewById(R.id.textView);
    tV.setText(msg);
}
```

*Intent is an object with key/value pairs, this is so that when switching between activities, you can use putExtra(key, value) as part of the intent, and then using getStringExtra() to get the value (where the value is the string)*

5. When does an inherited callback's super execute if its:
- a. Not overridden?  
Called whenever the function that it's overriding changes state
  - b. Overridden?  
Whenever the ordering of the function calls it.
6. Describe 4 tasks that code in an Activity's onCreate callback usually does.
- i) *Call super()*
  - ii) *inflate layout, set content view with resource reference to layouts*
  - iii) *Create references to views*
  - iv) *populate method with activity's data*

7. What is someVal?

```
long highScore =  
sharedPref.getInt(getString(R.string.saved_high_score) ,  
someVal) ;
```

*Default value if the getInt returns a null object*

8. What does getString do exactly?

```
long highScore =  
sharedPref.getInt(getString(R.string.saved_high_score) ,  
someVal) ;
```

*getString gets a string specified by the resource identifier which in this case is a string from strings.xml*

# Workshop Quizzes from some random thingy

1. A thin native client is to a hybrid app as a *Web Browser* is to a Mobile Web App
2. A phone manufacturer has customised android to block access to the GPS chips on some models of its product range.

So it can sell them more cheaply without undermining its more upmarket products.

Keeping the same hardware is more economical than manufacturing different hardware

What are the implications for each of the 3 types of Mobile apps? Why?

Web: Depends on the permissions of web browsers installed on the device.

Hybrid: GPS access Depends on the SDK permissions granted by the device

Native:

3. What java syntactic structures do the following terminate? Give as much as you can

a. Trailing ;

*End of a statement/line, e.g. Intent intent = new Intent();*

b. Trailing )

*Closure of the function parameter declaration or call e.g. myFunction() or myFunction(var1)*

c. Trailing }

*Closure of function, i.e. private void myFunction() { }*

4. When and how the API level is initially set on an emulator?

*The API Level of the emulator is set up during the creation of the emulator device, through the setup wizard.*

5. Can it be changed? How?

6. When and how is the API level initially set on the app you are developing?

*The API level initially set on the app is set during the Android project setup*

7. Can it be changed? How?

8. Consider an app compiled against API level 22 on a device running Nougat

a. This usually works. Why?

*As the Android OS is forwards-compatible, older APIs are supported on newer apps*

b. How could this go wrong?

*This can wrong due to security restrictions on the older APIs.*

c. Can it be fixed without changing app level or device levels? Explain.

9. Consider an app compiled against API 25 on a device running Lollipop

a. How could this go wrong?

b. If it goes wrong can it be fixed always or sometimes?

- c. Under what circumstances can it be fixed?

# Workshop Quizzes (2019)

## Week 2

### Question 1

- a) List two differences between activities and services in Android applications
- b) What is the relation between Content Provider and Content Resolver in Android applications
- c) List three components that are activated by Intents

	Harsil (100%)	Eric (100%)
a	<p>An <i>activity</i> represents a single screen with a user interface. Whereas a service is an application component that does not provide a user interface.</p> <p>An activity closes when another app is opened. Whereas a service can continue running in the background, even if that application is not opened.</p>	<ul style="list-style-type: none"><li>- Activities presents the user with the User Interface (UI)</li><li>- Services are the background process to make the logic behind the user interface (i.e. network API calls)</li></ul>
b	<p>The Content Resolver is a global instance in the application that provides access to your (and other applications') content providers. In other words, <code>ContentResolver</code> is the layer between the application and other applications's <code>ContentProviders</code>.</p>	<p>Relation between content provider and content resolver:</p> <p>Content Resolver is the single, global instance in your application that provides access to your (and other applications') content providers</p>
c	<p>activities, services, and broadcast receivers</p>	<p>Three components that are activated by intents</p> <p>Activities, Services, and Broadcast Receivers are activated by intents.</p>

## Question 2

Consider an app compiled against API level 27 on a device running Pie.

- a) This usually works. Why?
- b) How could this go wrong?
- c) Can it be fixed without changing the API level of the app and the device? Explain.

Harsil (83%)	Eric (67%)
<p>Pie is API 28</p> <p>a). This usually works as Android applications are generally forward-compatible with new versions of the Android platform. Because almost all changes to the android API are additive.</p> <p>b). This could go wrong, if the our application is using functions which are supported in API 27 but are, for some reason, not supported or deprecated in the newer version of Pie (API 28)</p> <p>c). User can workaround this issue by shipping the code for the function that is deprecated in the newer version of Android.</p>	<p>2. Consider an app compiled against API level 27 on a device running Pie.</p> <p>a) Pie runs on API 28, this works as Android is backwards compatible (backwards-compatibility) - meaning that older apps can run on newer devices.</p> <p>b) This could go wrong if a method/function/library used within the application becomes deprecated in API level 28, causing the functionality to not work (or even the app to crash)</p> <p>c) Not necessarily, a (3rd party) library can be installed - allowing the methods of that have been deprecated to be used.</p>



### Question 3

Assume you are developing an application that should be available in two different languages.

- a) Develop an application for each language separately. Do you agree??
- b) If not, what alternative approach would you recommended and why?

Harsil (100%)	Eric (100%)
<p>a). Developing two seperate applications for each language is not an ideal solution as it will be very resource intensive, in terms of both time and money.</p> <p>b). An alternative approach, that is widely adapted and highly recommended by the developer community, is to put all strings in one file (strings.xml) and reference the strings in that file in the application instead of hard-coding them. To port it in another language, we can simply create another xml file with the strings translated in other language.</p>	<p>a) No</p> <p>b) Using the strings.xml, and the resources.xml allowed a developer to createthe multiple strings that allow multi-lingual and regional support. For example:</p> <pre>&lt;resource type&gt;-b+&lt;language code&gt;[+&lt;country code&gt;]</pre> <p>allows the additional language codes and country codes to be added (though the language and country codes must be in the standard ISO format).</p>

## Week 3

### Question 1

- a) What would happen to the back stack if the user presses the back button?  
b) What is the difference between task and activity in Android applications?

Harsil (100%)	Eric (100%)
<p>a). The topmost activity will be destroyed and popped from the back stack. Furthermore, the previous activity is resumed and the previous state of its UI is restored.</p> <p>b). a task is the collection of activities that user interacts in an application. If user presses the home button, the task goes in background, loses focus, and all the activities in the task are stopped. The task can return to foreground and user can resume from where they left off.</p>	<p>a) The current activity is popped from the top of the stack and the activity is destroyed. Then the previous activity resumes with the previous state of its UI being restored</p> <p>b) A task is a collection of activities that users interact with when performing a certain job. While an activity is the views that the users can view.</p>

## Question 2

- a) List the callbacks (lifecycle and other) in sequence that occur for a reorientation event
- b) List the callbacks (lifecycle and other) in sequence that occur when the code executes finish() method
- c) Why there is a difference?

Harsil (96%)	Eric (64%)
<p>a). Assuming the application is already running and reorientation occurs, then the functions that are executed are as follows - onPause(), onSaveInstanceState(), onStop(), onDestroy(), onCreate(), onStart(), onRestoreInstanceState(), onResume()</p> <p>b). Assuming the application is running and finish() executes then these functions are executed - onPause(), onStop(), onDestroy()</p> <p>c). The Android system is smart enough to realise the intention of the user. When the reorientation occurs, Android knows that user wants to stay on the same activity that's why it will save the data, recreate the reorientated activity and restore its state. Whereas, if finish() is executed, it will destroy() the activity without saving anything.</p>	<p>a) onSaveInstanceState, onCreate and onRestoreInstanceState are invoked.</p> <p>b) It depends on the current state, assuming that activity is currently running onPause, onStop gets executed to stop the running of the activity then the method onDestroy is called when finish() runs.</p> <p>c) A device reorientation doesn't kill the activity, instead, it saves the states and restores the state of the instance.</p>

Question 3 (7+10+12=29 Marks)

a) What is the effect of calling `super()` in Lifecycle callbacks? support your answer with examples

b) What is the difference between `getPreferences()` and `getSharedPreferences()`?

c) Consider this piece of code:

```
SharedPreferences settings = getPreferences(0);
SharedPreferences.Editor editor = settings.edit();
editor.putString(param1,param2);
editor.commit();
```

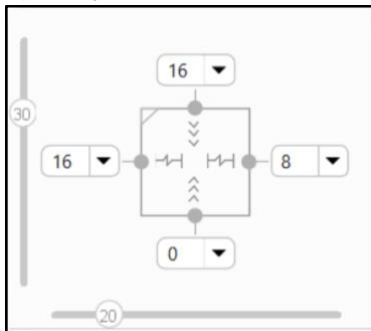
What are the data type and the role of **param1** and **param2**?

Harsil (100%)	Eric (100%)
<p>a). Consider the method <code>onBackPressed()</code> { <code>super.onBackPressed()</code>}. Here, the parent's <code>onBackPressed</code> method is called which is responsible for destroying the activity. If we comment the call to <code>super</code> method, pressing back will have no effect. Similarly, the <code>onRestoreInstanceState()</code> and <code>onSaveInstanceState()</code> call their respective parents' methods to restore and save bundles. If parent function calls are omitted, the app will not restore/save the state.</p> <p>b). <code>getPreferences()</code> gets the default preferences which is named after the app activity, whereas if we <code>getSharedPreferences()</code>, we can give a custom name for the file.</p> <p>c). <code>param1</code> is a string which is used uniquely save and retrieve the value, in this case <code>param2</code>. <code>param2</code> can be any string value that we intend to save with it's corresponding key (<code>param1</code>).</p>	<p>a) The method <i>super()</i> invokes the parent classes methods, (this is because when we override the method, we can't call the actual parents methods, e.g. we can get initialize the app) with the <code>onCreate</code> methods we can't call the parents activity initialisation method.</p> <p>b) <code>SharedPreferences</code> are preferences that are shared between the Activities while <code>Preferences</code> are just for its own activity.</p> <p>c) <code>param1</code> and <code>param2</code> are strings. Parameter 1 is the 'key' of the dictionary, while parameter 2 is the 'value' of the dictionary. Typically the editor can add any type into paramater depending on the type - in this instance we are adding a string with the <i>putString</i> method.</p>

## Week 4

Question 1 (12+4=16 Marks)

- Briefly describe all the numbers in the below picture.
- Why 20 is redundant?



Harsil (100%)	Eric (75%)
<p>a). 16 is the distance from top and left margins in "wrap content" and "match constraints" mode respectively. 8 is distance of view from right in "match constraint" mode. 0 is the distance from the bottom constraint i.e. it is touching another view/margin at the bottom, 30 is the vertical bias, whereas 20 is the horizontal bias.</p> <p>b). In match view, the view expands as much as possible to meet the constraints on each side, that's why 20 is redundant as there is no space left for the view to move.</p>	<p>a) Top and Bottom: 16 - Wrap Content: The view expands only as much as needed to fit its contents.</p> <p>Left and Right: 16 - Match Constraints: The view expands as much as possible to meet the constraints on each side</p> <p>20 - Horizontal Bias</p> <p>30 - Vertical Bias</p> <p>b) As left and right have different values and we're matching constraints as specified by the value, constraint bias has no valid input.</p>

Question 2 (4+5=9 Marks)

- a) List two differences between Styles and Themes.  
b) What is the main difference between Guidelines and Barriers in UI design?

Harsil (100%)	Eric (78%)
<p>a). Style is analogous to CSS for android. Style is collection of attributes used to specify look for views. Whereas theme is a style applied for an entire activity rather than an individual view.</p> <p>Secondly, styles allow you to define a set of attributes in one place and then apply them to multiple widgets. Whereas if theme is applied, every view in the activity/app applies those attributes that it supports.</p> <p>b). Guidelines have their own constraints from the margins of the app UI, whereas Barriers are constrained relative to the views of the app.</p>	<p>a) Styles are locally applied to the component (i.e. A style is a collection of attributes that specify the look and format for a View or window), Styles are inherited from the parent. The parent attribute in the &lt;style&gt; element lets you specify a style from which your style should inherit attributes</p> <p>Where as themes are global - A theme is a style applied to an entire Activity or app.</p> <p>b) Views are attached to the layout sides or horizontal and vertical guidelines. A barrier is an invisible view that contains reference to the views that you wish to use to form a “barrier” against</p>

Question 3 (7+4=11 Marks)

- a) What are the data type and the role of “param”?  
`registerReceiver(myBoardCastReceiver, new IntentFilter(param))`
- b) In order to listen to the new SMSs, you have to extend a class and implement a method. What are they?

Harsil (100%)	Eric (64%)
<p>a). It is of the type string. Its role is to specify the group of applications to which the information is to sent to ie. the application that are listening to that action string.</p> <p>b). You have to extend <code>BroadcastReceiver</code> and implement a method called <code>onReceive</code>.</p>	<p>a) <code>IntentFilter(String action)</code></p> <p>b) extending <code>BroadcastReceiver</code> and implement <code>onReceive()</code></p>

## Week 5

Question 1 (15+4 Marks)

- a) Briefly describe two ways to link a button to its onClick handler.  
 b) The method “**addListItem**” is intended to add three strings to the **listview** which is linked to the data source via an adaptor. There is a missing statement in the method which prevents from completing its task!! Code the missing statement.

```
private void addListItem() {
    adapter = new ArrayAdapter<String>(context this,
        android.R.layout.simple_list_item_1, listItems);
    listItems.add("First Value");
    listItems.add("Second Value");
    listItems.add("Third Value");
}
```

Harsil (95%)	Eric (100%)
<p>a). The two ways to link a button to its onClick handler are as follows:</p> <p>i). One of do it via the XML using attribute onClick</p> <p>ii). And another way is to do dynamically using java, using button.OnClickListener()</p> <p>b).adapter.notifyDataSetChanged();</p>	<p>1. You can assign a reference within the Activity handler using the <i>button.setOnClickListener()</i> using the view function or you can add a function to the button's manifest file.</p> <p>2. <code>adapter.notifyDataSetChanged();</code></p>

Question 2 (7+4=11 Marks)

- a) Assume you have a right to left drawer in you activity, write a java statement that opens it.  
 DrawerLayout drawer = findViewById(R.id.drawer\_layout);
- b) What is the effect of calling the below statement?  
 super.onBackPressed();

Harsil (82%)	Eric (45%)
<p>a). drawer.openDrawer(Gravity.RIGHT)</p> <p>b). It has effect of press of the back key. The default implementation simply finishes the current activity, but it can be overridden to do whatever you want.</p>	<p>1. <code>drawer.openDrawer(Gravity.START);</code></p> <p>2. It calls the superclass function (as you extended and overridden the method)</p>

Question 3 (4+4+4=12 Marks)

- a) Why return true after handling an options menu's menu item click?
- b) What is the main usage of the input parameter item for the method **onOptionsItemSelected**?

@Override

public boolean onOptionsItemSelected(MenuItem item)

- c) What is the difference between **onOptionsItemSelected** and **onNavigationItemSelected**?

92%

- a). To let the caller know that the event was finished successfully.
- b). *item* carries the information (for eg ID) of the item that was selected by the user. just like View carries information of the view (button) for button handlers.
- c). onOptionsItemSelected - called when an item in the options menu is selected.  
onNavigationItemSelected - called when an item in the navigation menu is selected.



## Week 6

### Question 1 (4 Marks)

a) This statement is suppose to replace a fragment and add it to the backstack, but there is something missing. Code the missing statement.

```
getSupportFragmentManager()  
    .beginTransaction()  
    .replace(R.id.frag1, new Fragment1())  
    .addToBackStack("f1");
```

100%

```
.commit();
```

### Question 2 (7+8 Marks)

a) Briefly describe the control flow (relation) between **onCreateViewHolder** and **onBindViewHolder** methods.

b) Briefly explain two differences between Fragments and Activities.

Harsil (100%)	Eric (47%)
<p>a). the onCreateViewHolder is called right when the adapter is created and is used to initialize the ViewHolder(s). onBindViewHolder method is called for each ViewHolder to bind it to the adapter. This is where we will pass our data to our ViewHolder. It is called by RecyclerView to display the data at the specified position.</p> <p>b). fragments live inside activities, and each activity can host multiple fragments. Like activities, they have a specific lifecycle, but unlike activities, they are not top-level application components. Advantages of fragments include code reuse and modularity (e.g., using the same list view in many activities). The main disadvantage is (some) added complexity. You can generally achieve the same thing with (custom) views in a non-standard and less robust way.</p>	<p>a)</p> <p>b) Fragments represents a behavior or a portion of user interface in an Activity. When an <b>fragment</b> is placed to the activity we have to request the instance to be saved by calling <b>addToBackStack()</b> during the <b>fragment transaction</b>.</p>

Question 3 (6 marks)

a) RecyclerView needs to have a subclass of RecyclerView.ViewHolder. Why?

Harsil (100%)	Eric (100%)
It needs to have a subclass so that the adapter can provide a binding from an app-specific data set to views that are displayed within that RecyclerView by called appropriate methods that are declared in the super class.	A ViewHolder describes an item view and metadata about its place within the RecyclerView. <a href="#">RecyclerView.Adapter</a> implementations should subclass ViewHolder and add fields for caching potentially expensive <a href="#">findViewById(int)</a> results.

## Week 7

### Question 1 (6+8 Marks)

- a) List the methods you **have to** implement when you extend the **CursorAdapter** class.
- b) Briefly explain the events that cause these two callbacks to get invoked.

@Override

```
public void onUpgrade(SQLiteDatabase db, int, i, int i1) { }
```

@Override

```
public void onDowngrade(SQLiteDatabase db, int, i, int i1) { }
```

86%

- a. List the methods you have to implement when you extend the CursorAdapter class.

```
public View newView(Context context, Cursor cursor, ViewGroup viewGroup)
```

```
public void bindView(View view, Context context, Cursor cursor)
```

- b. Briefly explain the events that cause these two callbacks to get invoked.

onUpgrade is Called when the database needs to be upgraded i.e. when saved database version is lower than the provided database version. Contrastingly, onDowngrade is called when the database needs to be downgraded.

### Question 2 (4+8=12 Marks)

- a) List the CRUD operations that need a **ContentValue** object as input to work.
- b) Describe the role and datatype of **param1** and **param2**.

```
ContentValues contentValues = new ContentValues();
```

```
contentValues.put(param1, param2);
```

100%

- a. List the CRUD operations that need a ContentValues object as input to work.

insert and update

- b. Describe the role and datatype of param1 and param2.

They are the key value pairs wherein param1 is the attribute (key) and param2 is the value of that attribute.

### Question 3 (4+12=16 Marks)

a) What is the effect of the statement in line **70**?

```
68.    customView.numberShapes = cursor.getCount();
69.    customView.shapes = shapes;
70.    customView.invalidate();
```

b) Describe the role and datatype of **param1**, **param2** and **param3**.

```
SQLiteDatabase db = getWritableDatabase();
result = db.delete(param1, param2, param3) > 0;
```

100%

a. What is the effect of the statement in line 70?

invalidate() find the best time and schedule for the next UI thread to execute. It queues the drawing on the thread so it can be carried out when the thread is not busy.

b. Describe the role and datatype of param1, param2 and param3.

param1 is the table name

param2 is the where clause, passing null will delete all rows.

param3 is where arguments

## Week 8

### Question 1

a) What is the main purpose of implementing the content provider in an Android application?

b) With respect to content provider, how does delete method serve multiple URIs?

85%

a). Content providers helps an application manage access to data stored by itself or stored by other apps, and provide a way to share data with other apps. It is an interface that connects data in one process with code running in another process.

b). We will be using the URIMatcher to match the type of the URI and patterns (for eg # for numbers) can be used to match multiple URIs of same pattern. Then URIMatcher can classify the relevant patterns as insert, delete, etc. Then we can have one switch statement to classify what sort of URI is the given one and take relevant steps to insert, delete, etc. the item from our database.

### Question 2

a) Explain this statement "CursorLoader's load is asynchronous"

100%

It is an asynchronous function as it might take a while for the "load" to, it will be independent from the current UI frame and will be running in a different thread. After it finishes loading the callback will be called. It is designed so that the application does not have to wait till the function finishes loading, which might introduce potential lags in the application

### Question 3

a) What are the roles and datatypes of param1 and param2

b) How did you declare and initialize object *resolver*?

```
resolver.insert(param1, param2);
```

92%

a).

param1: is the type Uri, which represents a URL of the table where data is to be inserted into.

param2: is ContentValues type, it represents the initial values for the newly inserted row.

b). `ContentResolver resolver = new ContentResolver(context);`

(where context is a variable of the type Context)

## Week 9

### Question 1

a) Briefly explain the role of the three input data types to the AsyncTask.

`private class GetLogo extends AsyncTask<String, Integer, Bitmap>`

80%

The three types used by an AsyncTask are the following:

1. **Params**, the type of the parameters inputted to the task when it is executed, in our case they are String.
2. **Progress**, the progress units published while it is performing the background computations, in this instance it is Integer.
3. **Result**, the datatype of result of the computation, for us it is BitMap.

### Question 2 (5+5=10 Marks)

a) What should you use to transform a LatLong point into a street address.

b) What is the meaning of "AsyncTask is a Generic class"?

100%

a). We can use the Geocoder instance's `getFromLocation` method to translate the passed in `LatLng` value to an `ArrayList` (addresses) of address objects returned by the Geocoder instance.

b). A *generic class* is the one which is parameterized over datatype of parameters. In other words, it allow one to write a generic class that will work with different types, allowing for code re-use.

Question 3 (4+4+4+4=16 Marks)

- a) How can an input parameter be accessed in `doInBackground`?
- b) When and on what thread does `doInBackground` execute?
- c) When and on what thread does `onPostExecute` execute?
- d) When and on what thread does `onProgressUpdate` execute?

100%

a). The input parameters passed in `doInBackground` is a *vararg*. They can be accessed as `varName[n]` in the method where `n` is the `(n + 1)`th parameter.

b). `doInBackground` is invoked on the background thread immediately after `onPreExecute()` finishes executing.

c). `onPostExecute` is invoked on the main thread after the `doInBackground` computation finishes.

d). `onProgressUpdate` is invoked on the UI thread after a call to `publishProgress()`. It's timing of the execution is not specific.

## Week 10

### Question 1 (8 Marks)

What is the difference between `event.getX()` and `event.getRawX()`? Support your answer with an example.

100%

`getRawX()` is guaranteed to return the absolute coordinates, relative to the device screen. Whereas `getX()` will return you coordinates, relative to the View, that dispatched them.

Say our layout has a simple button at the very centre of the layout. If we are getting the X and raw X values when the button is pressed, then `getX()` will get the coordinated relative to button's leftmost edge. Whereas `getRawX()` will get coordinates with respect to screen's left most edge.

### Question 2 (20 Marks)

If you have a frame layout with `id="frameid"`, write a piece of code that shows the X and Y of the click down event in a toast.

100%

```
View frame = findViewById(R.id.frameid);
frame.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        int x = (int) event.getX(); //use getRawX() and getRawY() if need the
        value from edge of the screen
        int y = (int) event.getY();

        int action = event.getActionMasked();

        switch (action) {
            case MotionEvent.ACTION_DOWN:
                String s = "Down X="+x+", Y="+y;
                toast(s);
                break;
        }
        return true;
    }
});
private void toast(String s) {
    Toast.makeText(this, s, Toast.LENGTH_LONG).show();
}
```



### Question 3 (8 Marks)

Briefly explain the role of the **getActionMasked's** output.

```
int i = motionEvent.getActionMasked();
```

100%

`getAction()` returns an event and a pointer id information. Whereas, `getActionMasked()` returns just an event (i.e., up, down, move) information. Other info is "masked" out.

## Week 11

### Question 1

- a) What is the difference between **onSingleTapUp** and **onSingleTapConfirmed** callbacks?
- b) What is the difference between **onDoubleTap** and **onDoubleTapEvent** callbacks?

a). `onSingleTapConfirmed` is executed when tap occurs. Unlike `onSingleTapUp`, this will only be called after the detector is confident that the user's first tap is not followed by a second tap leading to a double-tap gesture. `onSingleTapUp` is notified when a tap occurs with the up `MotionEvent` that triggered it. Source: [Android documentation](#)

b). `onDoubleTapEvent` is notified when a double-tap occurs. Whereas, `onDoubleTapEvent` is notified when an event within a double-tap gesture occurs, including the down, move, and up events. Source: [Android documentation](#)

### Question 2

Briefly compare between `onFling` and `onScroll` in terms of:

- a) Parameters
- b) Number of times the callback is called per gesture
- c) What the end-user had to do to invoke the callback

Credits: JY (He duxes all the units, so probably 100%)

a). `onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)`  
`velocityX` and `velocityY` are the velocities of this fling measured in pixels per second along the respective axes.

`onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)`  
`distanceX` and `distanceY` are the distances along the respective axes that have been scrolled since the last call to `onScroll` and this is NOT the distance between `e1` and `e2`.

b).

`onScroll` can be called one or more times during a gesture while `onFling` will only be called once at last triggered by the up motion event (`ACTION_UP` i.e., lifting the finger off the screen). `onScroll` is always invoked if the end-user performs a finger down followed by any move events, but `onFling` is only invoked if the move has the velocity with it.

### Question 3

- a) What is a convenience class?
- b) What problem does it solve?
- c) What's special about a convenience class for a listener interface?
- d) Why is it special?

JY	Eric (100%)
<p>a) A Convenience class generally means a class that doesn't do anything itself (i.e., doesn't have its own logic), but provides simplified access to other classes or groups of classes or interfaces.</p> <p>b) It resolves the problem that occurs when you need to provide the concrete implementation of the corresponding interface (e.g. <code>GestureDetector.OnGestureListener</code> which has six abstract methods to be implemented by the developer). Without Convenience class, we have to always implement all methods appearing in the interface definition and often the time, we don't use all of them but only interested in a few of them. In such case, implementing all methods would make your code a bit messy and less maintainable since later you may not be able to know which methods are the ones that contain the real business logic and which methods are dummy. By using Convenience class, this problem can be resolved perfectly since the Convenience class has already got its own implementation of all the methods from the interface definition and the developers only need to extend this class and override the methods they want to use.</p> <p>c) The default implementation of the methods in Convenience class for listener interface returns false for those methods with the return type as boolean.</p> <p>d) Because in the real implementation of the methods in the developer's code, they usually need to return true to indicate that the event has been consumed and handled or it may indicate that the user is interested in the subsequent sequence of the events (depending on the different situations, in our case, the return value for <code>onDown</code> method does have the meaning like this), so that the event that current view receives won't be propagated down to the view stack for further handling. The default</p>	<p>a) A convenience class generally means a class that doesn't do anything itself, but provides simplified access to other classes or groups of classes.</p> <p>b) Resolves the problems that occurs with calling anonymous class (e.g. <code>new View.OnClickListener()</code>). The activity needed to override all the method that occurs inside the interface definition (<code>View.OnClickListener</code>). Fortunately, usually it only have one methods inside it (<code>onClick()</code>). The convenience this problem, If the class extend the convenience class, then it just need to override the method it wants to use</p> <p>c) What's special about a Convenience class for a listener interface? returns false</p> <p>d) When you create a listener, it returns true to indicate that this method has been consumed(the event has been handled, you don't have to touch it anymore). So the API class specify it listener to false to prevent any block off on the apps such as the overridden listener in the developer's activity is skipped. (this can be called using <code>super()</code> )</p>

implementation returns false in order to avoid the case that the developers accidentally forget to override the methods in their code to avoid the unexpected behavior to happen.	
---	--

## Week 12

### Question 1 (12 Marks)

- a) What is the benefit of Instant Run?
- b) Describe three scenarios where it works and one where it doesn't
  - 1) Name the scenario.
  - 2) Under what conditions does it occur?
  - 3) What does Instant Run do in the scenario?

Credits: JY

a) Instant Run significantly reduces the time it takes to update your APK with code and resource changes.

b)

Hot swap: change implementation code of an existing method  
Instant Run pushes the changes without any restart.

Warm swap: change or remove an existing resource  
Instant Run pushes the changes and the activity restarts.

Cold swap: structural code changes  
Instant Run pushes the changes and the whole app restarts.

No swap: change the app manifest or change resources referenced by the app manifest or change an Android system UI element  
Instant Run doesn't run and the app has to be completely rebuilt and redeployed to the emulator or device.

### Question 2 (8 Marks)

How does ProGuard protect the source codes of Android Application? Briefly explain.

ProGuard manipulates Java bytecode the way you tell it with your configuration files and the rules they contain. ProGuard can do many things, including obfuscation, which is a process of creating source code in a form that is hard for a human to understand.

### Question 3 (4+5=9 Marks)

- a) What are the consequences if the private key of an app is lost but not compromised (i.e. no one else acquires it)?

b) What are the consequences if the private key of one of your apps becomes compromised without your knowledge (i.e. someone else acquires it)?

-2 possibilities

Credits: JY

a) The existing app on Google Play App Store can't be updated in the future. It would have to be a new app signed with a new key probably resulting in a loss of users.

b) You authoring identity and the trust of the user can be compromised. They could sign and distribute apps that maliciously replace your authentic apps or corrupt them. They could also sign and distribute new apps under your identity that attack other apps or the system itself, or corrupt or steal user data.

**END OF NOTES**