

Android Application Programming - Exam Notes

1. Service

Definition: A Service is a background component in Android that performs long-running operations without a user interface, like playing music or downloading files.

Stages / Lifecycle: onCreate() → onStartCommand() → onDestroy()

Steps to Use: 1. Create Service class → 2. Override lifecycle methods → 3. Register in Manifest → 4. Start service

Example:

```
public class MyService extends Service { public void onCreate() { Toast.makeText(this, 'Created', Toast.LENGTH_SHORT).show(); } public int onStartCommand(Intent intent, int flags, int startId) { Toast.makeText(this, 'Running', Toast.LENGTH_SHORT).show(); return START_STICKY; } public void onDestroy() { Toast.makeText(this, 'Stopped', Toast.LENGTH_SHORT).show(); } public IBinder onBind(Intent intent) { return null; } }
```

2. View

Definition: A View is the basic building block of UI in Android. Examples: Button, TextView, EditText.

Steps: 1. Create View → 2. Set properties → 3. Add to layout

Example:

```
Button button = new Button(this); button.setText('Click Me'); setContentView(button);
```

3. ViewGroup

Definition: A ViewGroup is a container that holds multiple Views or other ViewGroups and defines their arrangement.

Steps: 1. Choose ViewGroup → 2. Add child Views → 3. Set as content

Example:

```
LinearLayout layout = new LinearLayout(this); layout.setOrientation(LinearLayout.VERTICAL);  
Button b1 = new Button(this); b1.setText('Button 1'); layout.addView(b1); setContentView(layout);
```

4. Layout

Definition: Layout defines the structure and positioning of Views on the screen. Can be XML or programmatically.

Steps: 1. Create layout → 2. Add views → 3. Set as Activity content

Example (XML):

```
<LinearLayout xmlns:android='http://schemas.android.com/apk/res/android'  
    android:orientation='vertical' android:layout_width='match_parent'  
    android:layout_height='match_parent'> <TextView android:text='Welcome!'  
    android:layout_width='wrap_content' android:layout_height='wrap_content' /> <Button  
    android:text='Click Me' android:layout_width='wrap_content'  
    android:layout_height='wrap_content' /> </LinearLayout>
```

5. Adapter

Definition: An Adapter acts as a bridge between data sources and UI components like ListView or RecyclerView. Converts data items to Views.

Steps: 1. Create data → 2. Create adapter → 3. Attach to UI component

Example:

```
String[] fruits = {'Apple', 'Banana', 'Cherry'}; ArrayAdapter adapter = new ArrayAdapter<>(this,  
    android.R.layout.simple_list_item_1, fruits); ListView listView = findViewById(R.id.listView);  
listView.setAdapter(adapter);
```

Summary Table

Component	Definition	Example Use
Service	Background tasks without UI	Music or download service
View	Smallest UI element	Button, TextView
ViewGroup	Container for multiple Views	LinearLayout
Layout	Defines arrangement of Views	XML or Java layout
Adapter	Connects data to UI components	ListView with ArrayAdapter