📘 Android Development Notes for BCA 3rd Year

First Week Notes:

---

1. Introduction to Android

- Android is an open-source operating system based on the Linux **kernel**.
- It is mainly used for **mobile devices** (smartphones, tablets, etc.).
- Android applications are written in **Java, Kotlin**.

---

2. Creating a New Android Project

Steps:

1. Open **Android Studio** → `File > New > New Project`.
2. Choose a template (e.g., Empty Activity).
3. Configure project:
    - Name: `MyFirstAppOnGLA`
    - Package Name: `com.example.MyFirstAppOnGLA`
    - Language: Java/Kotlin
    - Minimum SDK: API 21 (Android 5.0)
4. Click **Finish** → Project is created.

---

3. Android Project Structure

```
MyFirstAppOnGLA/
 ├── manifests/
 │    └── AndroidManifest.xml  (App configuration)
 ├── java/
 │    └── com.example.MyFirstAppOnGLA
 │         └── MainActivity.java / MainActivity.kt (Code files)
 ├── res/  (Resources)
 │    ├── layout/ (XML layouts)
 │    ├── drawable/ (Images, shapes)
 │    ├── values/ (colors.xml, strings.xml, styles.xml)
 └── Gradle Scripts (Build configuration)
```

---

4. Activity in Android

- **Activity** = Single screen of the app (like a page in a book).
- Every app has at least one activity, usually **MainActivity**.

Example: `MainActivity.java`

```
package com.example.MyFirstAppOnGLA;

import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("Lifecycle", "onCreate called");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d("Lifecycle", "onStart called");
    }
```

```
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("Lifecycle", "onResume called");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d("Lifecycle", "onPause called");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d("Lifecycle", "onStop called");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("Lifecycle", "onRestart called");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("Lifecycle", "onDestroy called");
    }
}
```

---

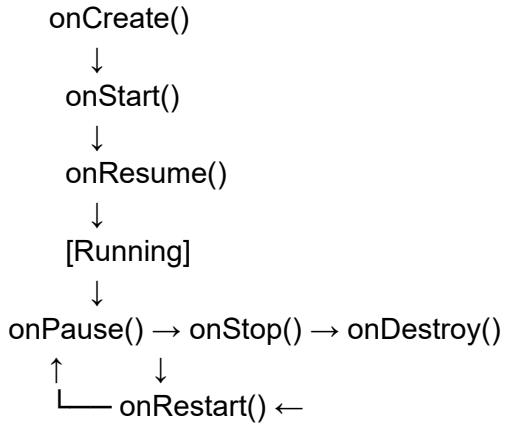
5. Activity Lifecycle

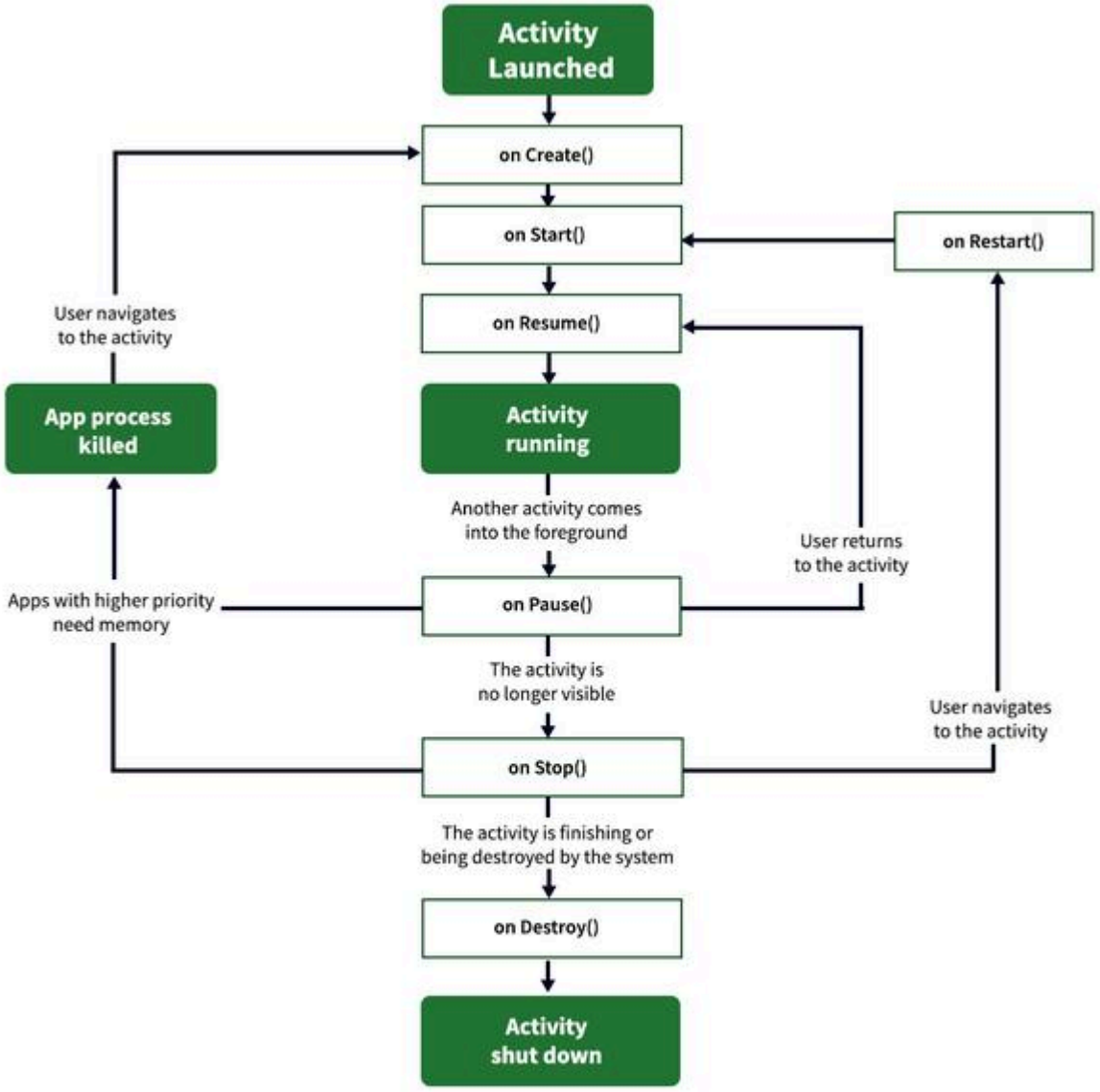
Lifecycle Methods:

1. `onCreate()` → Called when activity is created.
2. `onStart()` → Activity becomes visible.
3. `onResume()` → User starts interacting.
4. `onPause()` → Another activity comes in front.

5. `onStop()` → Activity not visible.
6. `onRestart()` → Called after activity is stopped and then restarted.
7. `onDestroy()` → Activity is destroyed.

Diagram:





Activity Lifecycle in Android

---

6. Layouts in Android

Layouts are XML files used to design UI.

Common Layouts:

1. LinearLayout

- Arranges elements **vertically** or **horizontally**.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello LinearLayout!" />
</LinearLayout>
```

2. RelativeLayout

- Positions views relative to each other or parent.

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello RelativeLayout!" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click"
        android:layout_below="@id/textView" />
</RelativeLayout>
```

3. ConstraintLayout

- Most powerful layout with flexible positioning.

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello Constraint!"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

---

7. Widgets in Android

Widgets = UI components that display or take input.

Common Widgets and Attributes:

1. TextView

```
<TextView
    android:id="@+id/tv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello TextView"
    android:textSize="20sp"
    android:textColor="@android:color/holo_blue_dark"
    android:textStyle="bold|italic"
    android:gravity="center"
    android:background="@android:color/holo_green_light"
    android:padding="10dp"
    android:ellipsize="end"
    android:singleLine="true" />
```

Important Attributes: `text`, `textSize`, `textColor`, `textStyle`, `gravity`, `background`, `padding`, `ellipsize`, `singleLine`

---

2. EditText

```
<EditText
    android:id="@+id/et"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your name"
    android:inputType="textPersonName"
    android:textColorHint="@android:color/darker_gray"
```

```
    android:maxLength="20"
    android:ems="10"
    android:singleLine="true"
    android:drawableStart="@android:drawable/ic_input_add" />
```

Attributes: `hint`, `inputType`, `textColorHint`, `maxLength`, `ems`, `singleLine`, `drawableStart`

---

3. Button

```
<Button
    android:id="@+id/btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click Me"
    android:background="@android:color/holo_green_light"
    android:textAllCaps="false"
    android:padding="12dp"
    android:textColor="@android:color/white" />
```

Attributes: `text`, `background`, `textAllCaps`, `padding`, `textColor`

---

4. ImageView

```
<ImageView
    android:id="@+id/img"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:src="@mipmap/ic_launcher"
    android:contentDescription="App Logo"
    android:scaleType="centerCrop"
    android:adjustViewBounds="true" />
```

Attributes: `src`, `contentDescription`, `scaleType`, `adjustViewBounds`

---

5. CheckBox

```
<CheckBox
    android:id="@+id/checkBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Accept Terms"
```

```
    android:checked="false"
    android:buttonTint="@android:color/holo_red_dark" />
```

Attributes: `text`, `checked`, `buttonTint`

---

6. RadioButton & RadioGroup

```
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <RadioButton
        android:id="@+id/radio1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 1"
        android:checked="true" />

    <RadioButton
        android:id="@+id/radio2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 2" />
</RadioGroup>
```

Attributes: `text`, `checked`, `buttonTint`, `orientation` (for group)

---

7. ProgressBar

```
<ProgressBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:indeterminate="true"
    android:visibility="visible"
    android:progress="50"
    android:max="100"
    style="?android:attr/progressBarStyleHorizontal" />
```

Attributes: `indeterminate`, `visibility`, `progress`, `max`, `style`
