LAB Exercise 1

- Create an activity named **LifeCycle**.
- Implement all the lifecycle methods (e.g., onCreate(), onStart(), onResume(), etc.) and log a message in each callback method.
  For example, in the onCreate() method, use:

  Log.d("LifeCycle Activity", "onCreate called")
- Run the app and observe the logs via **Logcat**.
- Perform various actions such as switching between apps, rotating the device, and pressing the back button.
- Observe how the lifecycle methods are called and in what order.

LAB Exercise 2

- Create an activity named **Greeting**.
- Add the following UI components in your layout file:
  - An EditText with the ID **name**
  - A Button with the ID **button**
  - A TextView with the ID **hello**
- **Functionality:**
  - When the user enters their name in the EditText and clicks the Button, the TextView should display a greeting like:

Hello, [Your Name]

- **Note:**
  Implement the button click listener using OnClickListener **in the Activity code**. Do **not** use the onClick attribute in XML.
- **Problem:**
  - When the screen is rotated, the greeting text in the TextView is lost.
  - **Fix this issue** by preserving the data during configuration changes.

LAB Exercise 3

- Enhance **Lab Exercise 2** by adding an **Options Menu**.
- Add a menu item named **Send**.
- When the **Send** item is clicked, retrieve the text from the EditText and pass it to another activity using an **Intent**.

LAB Exercise 4

- Create an activity named **PickActivity**.
- Add an ImageView to the layout with the dimensions **300x150dp**.
- Set an **onClick listener** on the ImageView to allow users to pick a photo from the gallery.
- When a photo is selected, display it in the ImageView.

Note:
When demonstrating how to pick an image from the gallery, use **both** startActivityForResult and the **Activity Result API**.
Make sure to **understand and explain the differences** between them.

LAB Exercise 5 ***(To Be Done Together)***

- Create an activity named **ListActivity**.
- The goal is to display a list of contacts using a **ListView** component.
- A ListView is a view group that displays items in a vertically scrollable list.
- Create a separate **Kotlin data class** to hold contact information.
- Bind the list of contacts to the ListView in your ListActivity.