**Lab Exercise 9- Toast Messages in Jetpack Compose**

**Objective:**

Learn how to display a simple **Toast** message in **Jetpack Compose** when a button is clicked.

**Prerequisites:**

* Basic understanding of **Kotlin** programming.
* Familiarity with **Jetpack Compose** and setting up a Compose project in Android Studio.

**Steps:**

**Step 1: Set up a new Jetpack Compose project**

1. Open **Android Studio**.
2. Create a new project by selecting **New Project** -> **Empty Compose Activity**.
3. Set the project name and finish the setup.
4. Ensure that **Jetpack Compose** is enabled in your project. Your build.gradle should have the following dependencies:

dependencies {

implementation "androidx.activity:activity-compose:1.7.0"

implementation "androidx.compose.ui:ui:1.4.0"

implementation "androidx.compose.material:material:1.4.0"

implementation "androidx.compose.ui:ui-tooling-preview:1.4.0"

}

**Step 2: Create a Toast in Jetpack Compose**

1. Open your MainActivity.kt file.
2. Modify the MainActivity class to display a button that will trigger a **Toast** message when clicked.
3. Use the Toast.makeText() function inside the onClick lambda of a button.

Here's an example:

import android.os.Bundle

import android.widget.Toast

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.layout.Column

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.padding

import androidx.compose.material.Button

import androidx.compose.material.Text

import androidx.compose.material3.MaterialTheme

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import androidx.compose.ui.unit.dp

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.platform.LocalContext

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

MyApp()

}

}

}

@Composable

fun MyApp() {

// Get the current context

val context = LocalContext.current

Column(modifier = Modifier.fillMaxSize().padding(16.dp)) {

Button(onClick = {

// Show Toast message

Toast.makeText(context, "Hello from Jetpack Compose!", Toast.LENGTH\_SHORT).show()

}) {

Text(text = "Show Toast")

}

}

}

@Preview(showBackground = true)

@Composable

fun DefaultPreview() {

MyApp()

}

**Step 3: Run the Application**

1. Run the app on an emulator or a physical device.
2. Click the button labeled **"Show Toast"**.
3. A Toast message will appear at the bottom of the screen with the text **"Hello from Jetpack Compose!"**.

**Explanation:**

1. **LocalContext.current**: This provides access to the Context from within the Jetpack Compose @Composable function. The Context is required to display a Toast.
2. **Button Composable**: We use a Button composable that triggers the Toast when clicked. The onClick parameter specifies what happens when the button is pressed.
3. **Toast.makeText()**: This is the traditional way of showing a Toast in Android, and it still works within Jetpack Compose as long as you provide the right Context.

**Lab Exercise Task:**

1. Modify the message displayed in the Toast to include the current time.
   * Hint: Use SimpleDateFormat or java.time.LocalTime.now() to get the time.
2. Add another button that displays a different Toast message when clicked.

**Example of using time in a Toast:**

Button(onClick = {

val currentTime = java.time.LocalTime.now()

Toast.makeText(context, "Current time: $currentTime", Toast.LENGTH\_SHORT).show()

}) {

Text(text = "Show Time")

}

This lab exercise teaches students how to integrate traditional Android UI elements (like Toast) with Jetpack Compose, focusing on how they can handle context and events in Compose.