**Google CodeLab: Jetpack Compose Basics**

**INDIVIDUAL WORK**

In this lab, you will understand how to design a UI with Jetpack Compose.

You will work on this codelab during the session.

<https://developer.android.com/codelabs/jetpack-compose-basics>

Log yourself in with your Gmail on the top right

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Answer the following questions and submit in Classes.

1. What and why do we use @composable?

@composable is a keyword used to mark composable functions in a compose app, which can be used to call other composable functions. This enables function to call other @composable functions. This annotation tells the Compose app to add special support to the function for updating and maintaining UI of the app over time. It lets the structure code of the app into small chunks. By making small reusable composable, it’s easy to build a library of UI elements used in app.

1. What is Surface?

It is a UI element that accepts an optional modifier parameter, so that modifiers inside surface element tell a UI element how to layout, display or behave within its parent element. The components nested inside Surface will be drawn on top of that background color. Material components defined in surface are built to make experience of the app better by taking care of common features such as choosing an appropriate color for text. This material component in the Compose app is built on top of the other fundamental components.

1. Why would you use Preview?

We use @Preview annotation before parameterless Composable function or functions with default parameters to use Android Studio preview. It will create a preview of a composable function in Android Studio without running an app on emulator or android device.

1. How to add components such as Text and Button?

Button is a composable provided by the material3 package.

First import androidx.compose.material13.Button. Then define a composable function and add Button composable to it with other parameters such as onClick() which takes care of the logic which will be invoked when button is clicked. We can also add Text to it and use it in other composable functions.

To add Text, we have to import androidx.compose.material3.Text. Then define a composable function which takes string parameter as input. In this function, we can pass string parameter to Text(text: String) method. This composable functions can be used in other composable functions.

1. How to create columns and rows in Kotlin?

Column and Row are composable functions that take composable content. First import import androidx.compose.foundation.layout.Column or import androidx.compose.foundation.layout.Row. Then we can add composable ‘Column’ or ‘Row’ with other parameters such as text or other modifiers. Child inside Columns are placed vertically while child inside Rows are placed horizontally.

1. What is the state of a component in Compose?

State of a component is a interface that holds some value and trigger UI updates (recompositions) whenever that value changes.

1. What is a lazy list?

Lazy list is a list which uses ‘LazyColumn’ and ‘LazyRow’ to display scrollable column or row in a list. LazyColumn or LazyRow renders only the visible items on screen, allowing performance gains when rendering a big list.

1. What is/are the differences between Android Java way to build UI and Jetpack Compose for Kotlin?

In Android java way to build UI, we use XML to define layout files and use java to add logic to the UI components. This XML and java files are defined separately in different folders. In Jetpack Compose UI and logic of the UI component are defined in a single file using Kotlin.

1. Go to your profile and provide a screenshot of your profile.

<https://developers.google.com/profile>

Click MyProfile

Provide a screenshot that shows how many Codelabs you completed so far.

A screenshot of a computer

Description automatically generated