

# **BCS 371 Lab – Hello World (Android Studio)**

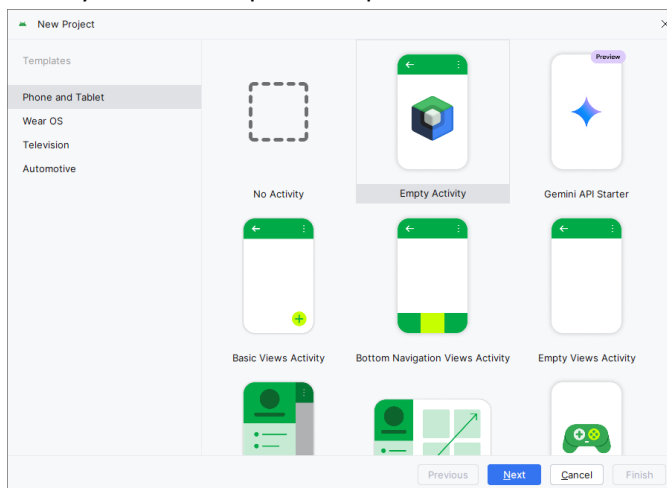
## ***Overview***

In this lab you will create an app that displays text. You will run it on an Android Virtual Device (AVD).

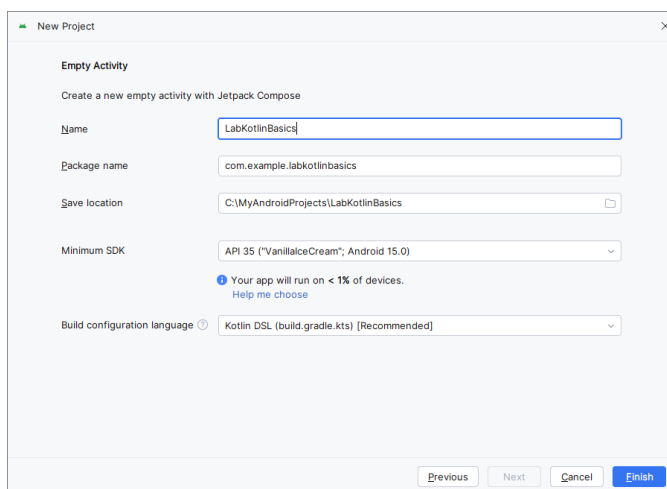
## ***Create Android Studio Project***

Create a new Android Application Project in Android Studio.

1. Go to File|New|New Project. Choose Empty Activity and press Next. This will create an empty activity that uses Jetpack Compose.

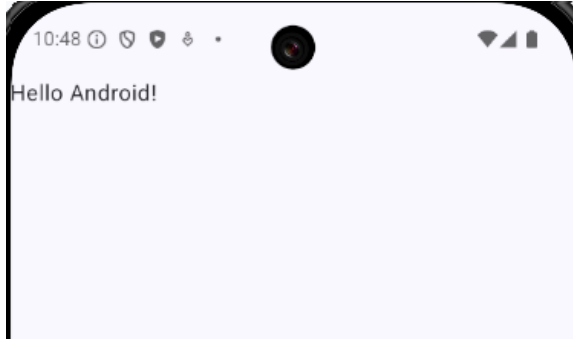


2. Choose a project name, location to store the project, language (Kotlin), and the minimum API level. Click Finish when done.



## ***Run the app***

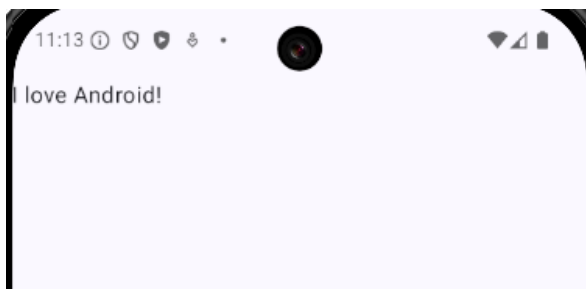
Go to Run | Run 'app'. The emulator should start and the app will be running inside of it. It should say Hello Android! in the app.



## ***Run Simplified Code***

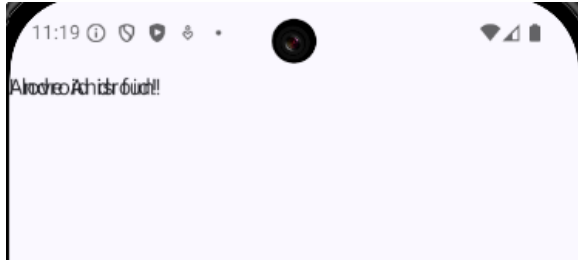
Update the app as follows:

1. Open the **MainActivity.kt** file. It is located in the **app/java/com.example.helloworld** directory.
2. Find the Greeting method and delete it.
3. Find the GreetingPreview method and delete it.
4. Inside the onCreate method, find the setContent block. Inside this block find the Scaffold and remove the call to Greeting and replace it with the following:  
`Text(text="I love Android!", modifier=Modifier.padding(innerPadding))`
5. Run the app.



## ***Add New Text***

Add another Text element to the setContent block that displays the message "Android is fun! ". It will not display what is intended though (make sure to pass in the modifier value as the second parameter to Text). Here is what it will look like:

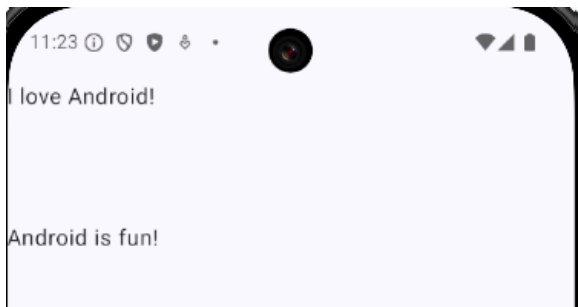


## ***Add Column Element***

Surround the two Text elements with a Column block. For example:

```
setContent {  
    Column {  
        Text(text = "I love Android!")  
        Text(text = "Android is fun!")  
    }  
}
```

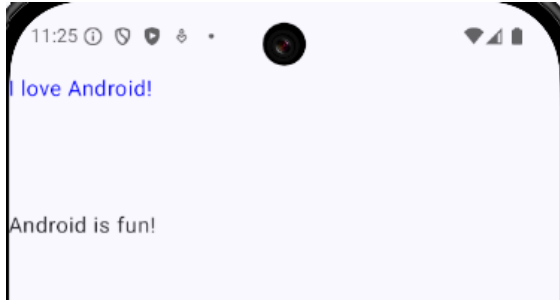
Here is what it should look like:



## ***Change Text Color***

Change the color of the first Text element to blue. The Text element takes a comma separated list of arguments. Add the following as the last parameter inside the parenthesis: `color = Color.Blue`

Here is what it should look like:

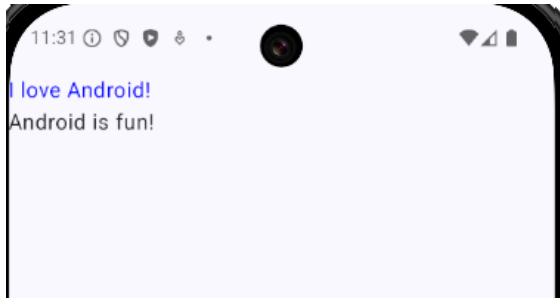


## ***Apply innerPadding Modifier to Column***

Apply the innerPadding modifier to the Column instead of the two Text elements.

- The Column should look like the following:  
Column (modifier = Modifier.padding(innerPadding))
- Remove the modifier parameter from both Text elements.

Here is what it should look like:

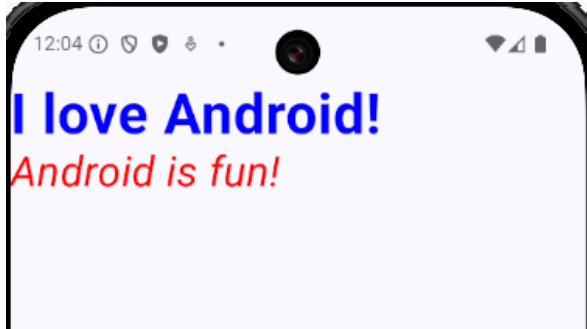


## ***Change Multiple Text Properties***

Make the following changes:

- Change the color of “Android is fun!” to red.
- Change the fontSize of the first Text element to 40.sp.
- Change the fontSize of the second Text element to 30.sp.
- Make the fontWeight of the first Text element bold.
- Make the fontStyle of the second Text element to italic.

Here is what it should look like:

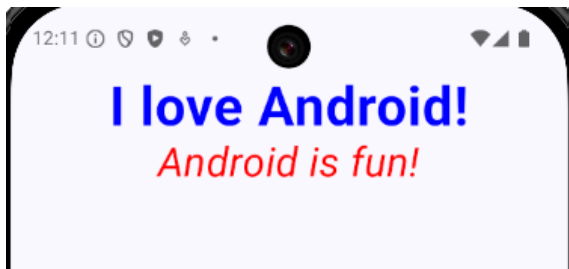


## ***Center the Elements in the Column***

Make the following updates to the Column element:

- Add parenthesis to Column: `Column( ) { // other stuff here... }`
- Add the following inside the parenthesis:
  - Change the Column modifier parameter so that it uses `fillMaxSize` too. It should look like the following: `Modifier.padding(innerPadding).fillMaxSize()`  
`modifier=Modifier.fillMaxSize()`
  - Add a `horizontalAlignment` argument that will center the column items:  
`horizontalAlignment = Alignment.CenterHorizontally`

Here is what it should look like:



## ***Create and Use a String Resource***

1. Open the **strings.xml** file. It is located in the **app/res/values** directory.
2. Add the following element to `<resources>`:  
`<string name="android_message_1">I love Android SR!</string>`
3. Update the first Text element in code.  
 Replace: `text = "I love Android!"`  
 With: `text = stringResource(id = R.string.android_message_1)`

Here is what it should look like:

