CSCI 448 – Lab 01A
Wednesday, January 18, 2023
LAB IS DUE BY **Tuesday, January 31, 2023 11:59 PM**!!

There are two main goals for today's lab:

1. Create string resources to use as labels in our app
2. Create a custom composable UI

This will be an app that lists your favorite collegiate monsters. The next set of labs will get you familiar with the different components of an app (resources, layout, activity) and comfortable with a bit of Kotlin code.

Step 0 – Create the Monster Lab Project

If you hadn't done so with Lab00A, create a project named Monster Lab. Be sure to target Android 10 and start with a Compose Activity (Material3). If you deploy your app at this point, you should see the generic "Hello Android" starting point. This matches with the result of Lab00A.

Step 1 – Copy The Drawables Over

Included with this lab writeup is a drawable folder. This contains a set of images we'll leverage in our app. Copy the contents into the project's `app/src/main/res/drawable` folder.

Step 2 – Build A Monster Card!

Part 2.I – Make The Strings

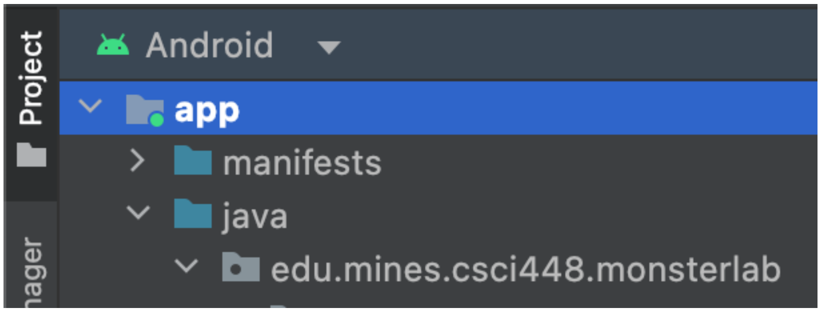
There are several labels we'll need to create and we'll place those in the `strings.xml` resource file we'll soon grow accustomed to. Create the following strings with the corresponding names and values:

String Name	String Value
name_monster_mike	Mike Wazowski
description_monster_mike	Eager to scare!

Part 2.II – Make The Monster Card

We'll start forward thinking to our file structure and organization scheme. Let's get into the habit of placing files into packages that dictate the file's usage.

Make sure you are in the Android view for the project tab. Right click on the existing package name under the java folder and in the popup menu select New > Package. Name the new package `monsterlab.presentation`.



Now in this new presentation package, create a new Kotlin File and name it `MonsterCard.kt`. This will display the information pertaining to a specific monster.

Create our initial composable:

```
@Composable
fun MonsterCard() {

}
```

We'll also create our preview of the composable so we can see what it is we're building.

```
@Preview
@Composable
fun PreviewMonsterCard() {
  MonsterCard()
}
```

If you're in the split view, you should now see your code and the preview of the UI (which is blank). Let's build up our composable!

Our composable tree will look like the following:

- `MonsterCard`
    - `Card`
        - `Row`
            - `Image`
            - `Column`
                - `Text`
                - `Text`

Let's now add each element in order.

Create a `androidx.compose.material3.Card` composable and set the `padding` modifier to `8.dp`.

Nest inside the trailing lambda a `Row` composable with the following modifier properties set:

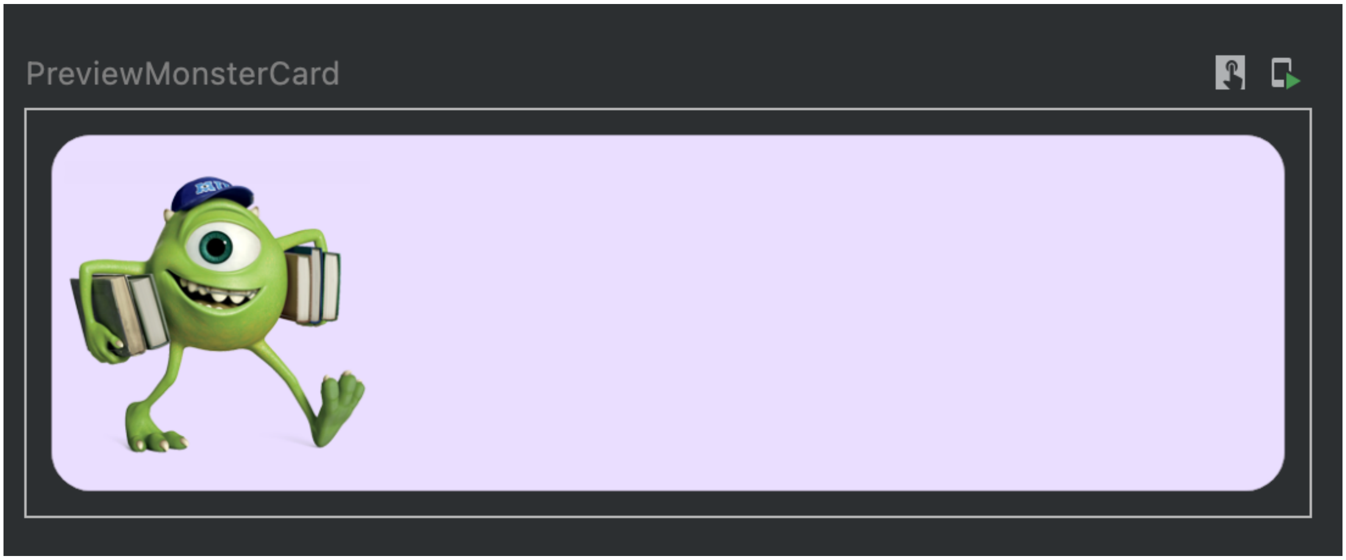
- `background(color = MaterialTheme.colorScheme.primaryContainer)`
- `fillMaxWidth()`
- `padding(top = 8.dp, bottom = 8.dp, start = 4.dp, end = 4.dp)`

Continuing to nest in the next trailing lambda, add an `Image` composable setting the following parameters:

- `painter = painterResource(id = R.drawable.monsters_university_character_young_mike_icon)`
- `contentDescription = stringResource(id = R.string.name_monster_mike)`

The two resource commands perform our lookup and resolution of the drawable and string resources that correspond to the associated names.

If you refresh the preview, you should now see Mike walking across your screen.

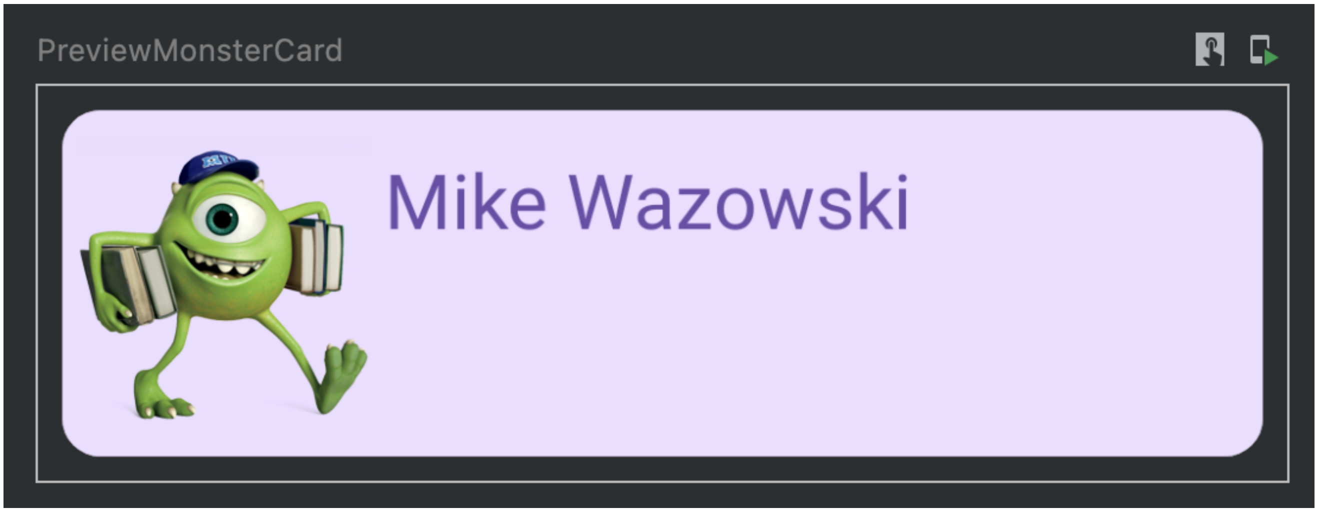


Continuing in the `Row` lambda, after the `Image` we'll now add a `Column` composable. Set the `padding` to be `4.dp` for the `Column`.

We'll add our first `Text` composable inside the `Column`. Set the following parameters on the `Text`:

- `text = stringResource(id = R.string.name_monster_mike)`
- `fontSize = 28.sp`
- `color = MaterialTheme.colorScheme.primary`

Refreshing the preview again, we'll now know who we're looking at!

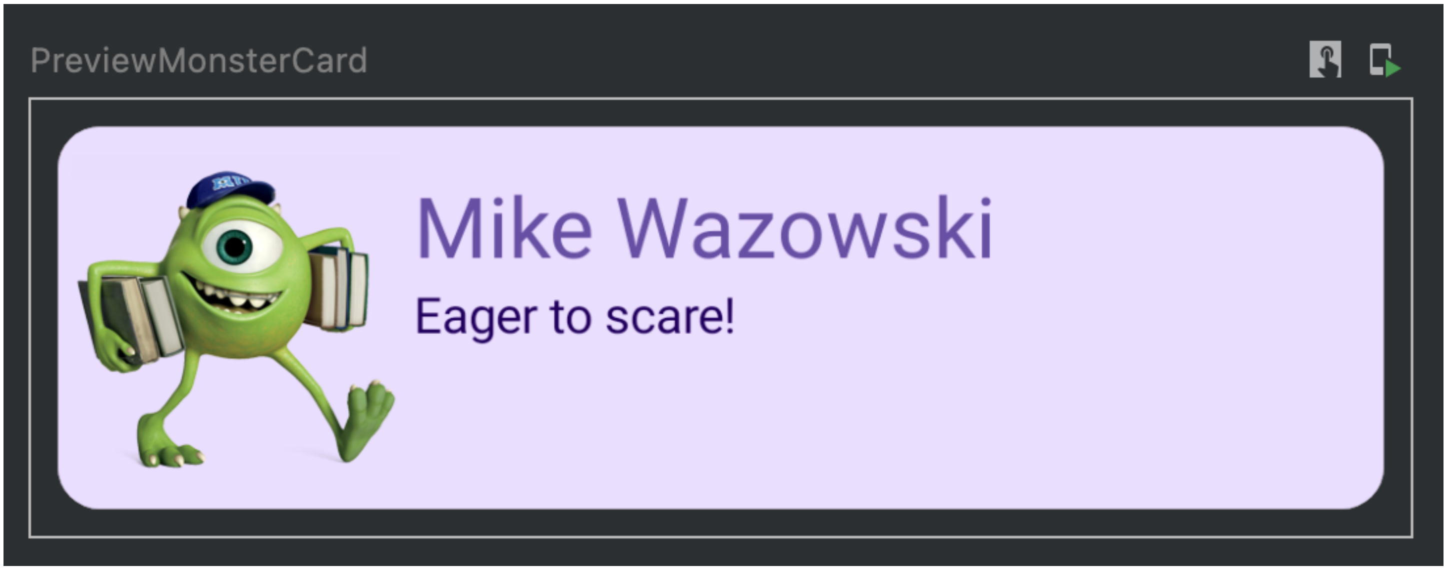


Mike Wazowski

We're now ready to add the final Text composable inside the Column. Set the following parameters on the second Text call:

- text = stringResource(id = R.string.description_monster_mike)
- fontSize = 14.sp
- color = MaterialTheme.colorScheme.onPrimaryContainer

When your preview matches the following image, continue to Part 2.III.



Mike Wazowski

Eager to scare!

Part 2.III – Activity – MainActivity.kt And The MonsterLabScreen

We will spend time discussing Activities as they are the life blood of our app. For now, we'll treat it as the brain of our app where we connect everything of what it does. Think of it as the equivalent to main() in our C++ or Java programs (that is not wholly accurate, but does reflect how the app works at this point int time).

For our purposes, the onCreate() function is called when the app starts and the activity is created. Here is where we specify which composable to use as our UI. This is done through the

```
setContent {
  MonsterLabTheme {
    Surface( ... ) {
      ...
    }
  }
}
```

lambda. We are setting the content to be the Surface and Text composables. We need to update it to use our custom MonsterCard. Remove the existing Greeting composable that is given to us.

As we will want to begin thinking about our UI as a set of screens, create a new composable function called MonsterLabScreen and have it call our custom MonsterCard composable.

```
@Composable
fun MonsterLabScreen() {
  MonsterCard()
}
```

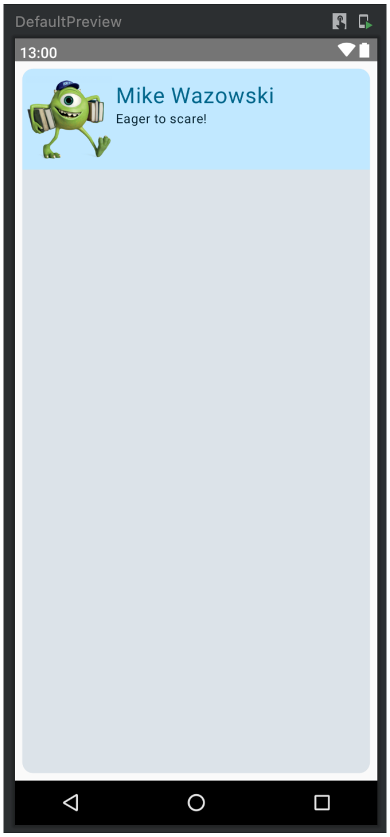
In the `MainActivity setContent`, call your new function instead of `Greeting`..

```
class MainActivity : ComponentActivity() {
  override fun onCreate(savedInstanceState: Bundle) {
    super.onCreate(savedInstanceState)
    setContent {
      MonsterLabTheme {
        Surface(...) {
          ~~Greeting("Android")~~
          MonsterLabScreen()
        }
      }
    }
  }
}
```

Likewise, in the existing `Preview` composable, call `MonsterLabScreen` in place of `Greeting`. We'll also add an argument the preview to show the full device screen.

```
~~@Preview~~
@Preview(showSystemUi = true)
```

The code preview will now show us Mike on our device! (Don't worry about the gray `Card` background having a height the same size as the screen. That will be resolved in the next lab.)



Mike Wazowski
Eager to scare!

Step 3 – Deploy Your App

Once you have the screen composable set, deploy your app to your device. While the colors may differ (based on light/dark theme), you should now have Mike Wazowski on your device!

When Lab01A is complete, continue on to Lab01B to add more monsters to your monster lab.

LAB IS DUE BY **Tuesday, January 31, 2023 11:59 PM**!!