KCL Tech: Build X: Android – LogInTheProgress

This is the final week of my part of the Android fundimentals part of this Android workshop. From next week, we will be building on top of what we have learnt to build a simple Youtube clone. In this workshop therefore we will be tying everything together.

Recap Challenges

a) Create and run a new Empty Android Project. Call the activity LoginActivity.
b) Add 2 EditTexts and one Button vertically
d) Set the background colour

Challenge 1: Create another Activity

a) Run the app
b) See all the new files that have been created
c) Look at the AndroidManifest

Up to now we have been working on one screen but most applications need more then one screen. Therefore to create a new screen i.e. Activity we go on File > New > Activity > Empty. We can give it a name; I called mine Main Activity; then click Finish. This will generate a new Java file for the activity and a new blank XML layout.

If you look in the AndroidManifest, you can see new <activity> tags for the new Acitivity but does not include the launcher tags inside. This is because we do not want this activity to show up on the phone's app launcher.

If an Activity is not on the AndroidManifest then it can not be opened. Try running the app, with and without the new acitivity's tags.

Challenge 2: Launch Activity

a) Create a new Intent
b) Send information for one Acitivty to another
c) Start Activity

Although we've created a new Activity, we can't yet see it. This is because it has not started it in the app. To do so, we use something called an intent. An intent can be used for a range of different things like sharing a photo or send a message. We will be using it to tell Android that we want to start another Acitivity by passing the current activity class and the class of the activity we want to start. Then pass the intent into a function *startActivity* In our case it will be…

```
Intent intent = new Intent(LoginActivity.class, MainActivity.class);
startActivity(intent);
```

Challenge 3: Check username and password

In the Java code for the LoginActivity, we will be giving our app the functionality of a login app. The user can enter a username and password which then logs them into the main app. To do this, we will need to implement a Button listener so that when the button is clicked, it get the username and password checks them. For this example just use *if* statements and compare the text to a hard-coded using the function string.*equals("username").* The application should now only allow the user to enter the MainActivity if the user enters the correct username and password.

NOTE: This implementation is only a simplified version. We would normally not store the username and password in plain-text.

Challenge 4: Display the logged in username in MainActivity

Finally we want to display the username of the user that logged in so that it shows up in the TextView in MainActivity. To do this we need to make a few modifications to the code. As the username is only entered in the LoginAcitivity and not saved anywhere MainActivity can access, we will need to send it across when we start the Activity. Intents allow us to do this, but putting an *Extra value e.g. intent.putExtra("username", "value");. Extras* can pass Strings, integers and booleans between activities.

To get the Extra from the MainActivity, you will first need to get the Intent that started it using Intent intent = *getIntent();* then get the extra values from it using *Bundle extras = getExtras();.* A Bundle is just a class to hold all the Extra pairs. After checking that the bundle is not null, we can get the Extra value from the bundle use the function *extras.getString("value") and then set it as the value for the TextView.*