BCS 371 Lab – Kotlin Basics

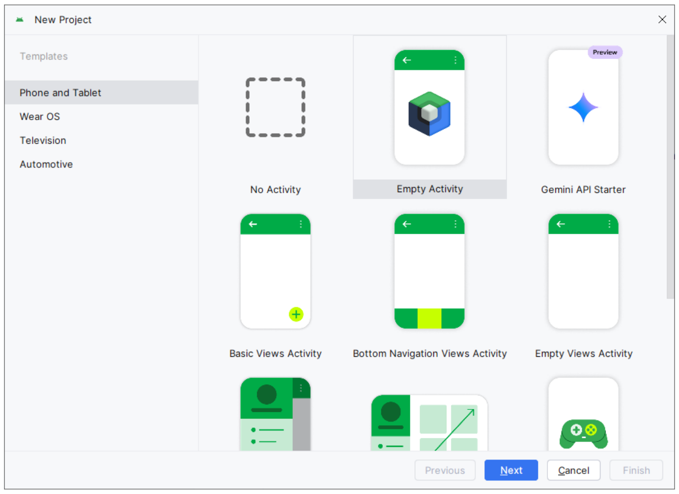
Overview

Write basic Kotlin code in an Android app. The app will use the Logcat window for output (not the GUI).

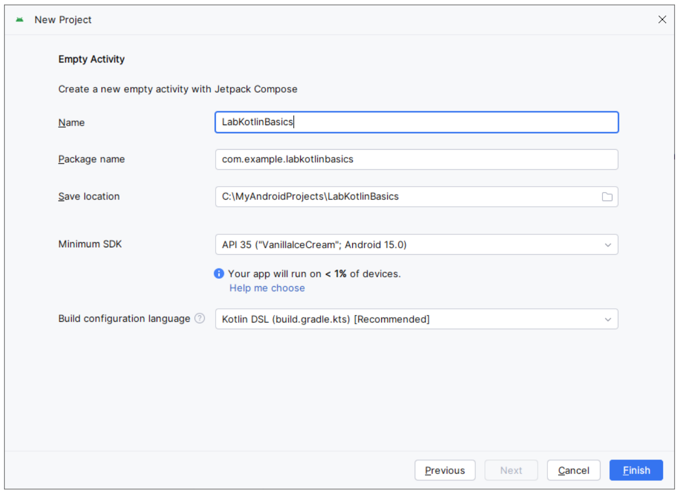
Create Android Studio project

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.



New Project

Empty Activity

Create a new empty activity with Jetpack Compose

Name: LabKotlinBasics

Package name: com.example.labkotlinbasics

Save location: C:\MyAndroidProjects\LabKotlinBasics

Minimum SDK: API 35 ("VanillaIceCream"; Android 15.0)

Your app will run on < 1% of devices.
Help me choose

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

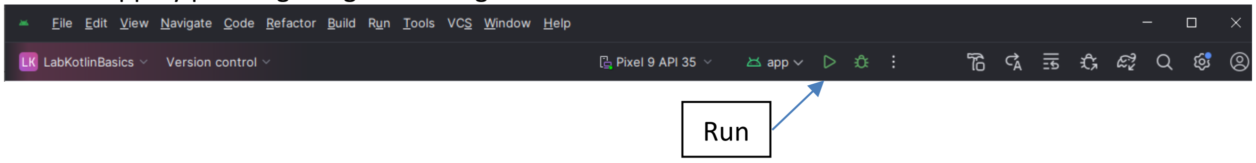
Cleanup project start code

Do the following in MainActivity.kt:

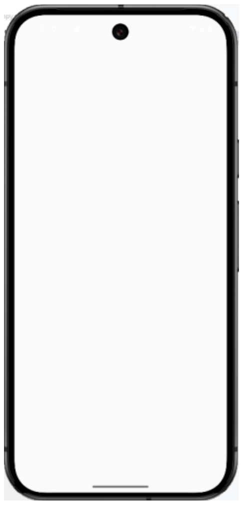
- Remove the Greeting and GreetingPreview functions from the MainActivity class.
- Remove all code from the onCreate function except for the call to super.onCreate.

Run the app

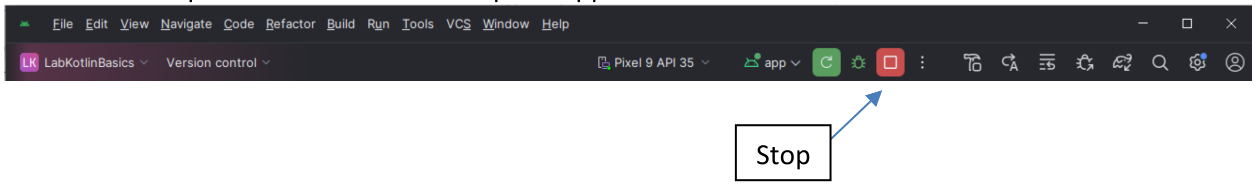
Run the app by pressing the green triangle in the toolbar.


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
LK LabKotlinBasics
Version control
Pixel 9 API 35
app
Run

It should run in the emulator and there will be nothing on the screen. Here is a screenshot:



Press the red square in the toolbar to stop the app.

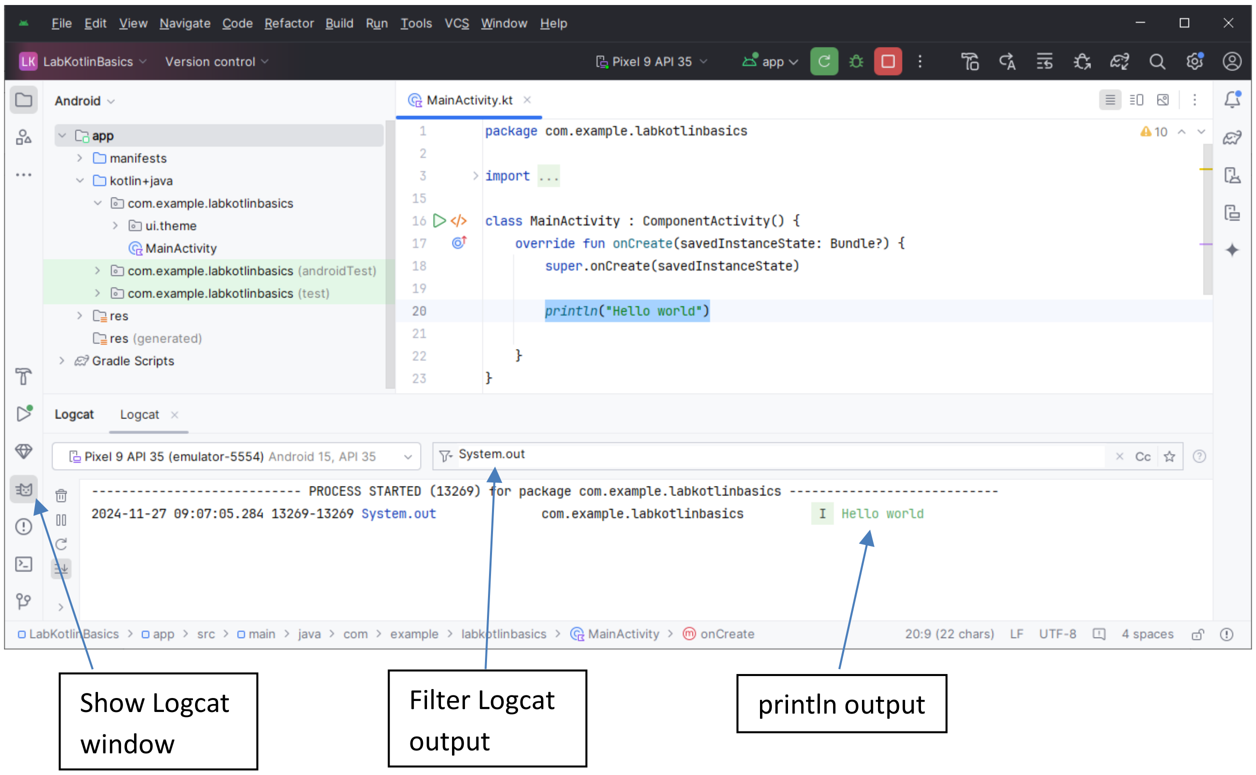

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
LK LabKotlinBasics
Version control
Pixel 9 API 35
app
Stop

Print message in Logcat Window from MainActivity.onCreate

Print a "Hello world" message in the Logcat window. Use the following statement to print to the Logcat window (put this code after the call to super.onCreate):

println("Hello world")

Run the app and check for the "Hello world" message in the Logcat window. Type in "System.out" to filter the output in the Logcat so you can easily see your message. Here is a screenshot:



```kotlin
package com.example.labkotlinbasics

import ...

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        println("Hello world")

    }
}
```

```
---------------------------- PROCESS STARTED (13269) for package com.example.labkotlinbasics ----------------------------
2024-11-27 09:07:05.284 13269-13269 System.out              com.example.labkotlinbasics          I  Hello world
```

Clearing Logcat. You can clear the Logcat window messages by right-clicking inside the Logcat window and choosing Clear Logcat from the context menu.

Declare and use a variable

Declare a string variable named message and put the value "Hello world" inside of it. Update the println statement to use the message variable instead. Run the app and check the output.

Use a for loop to calculate a sum

Do the following:

- Declare a sum variable and initialize it to 0.

- Write a for loop that runs from 1 to 5. Inside the for loop add the current value of the loop control variable to sum.
- After the loop print the sum in the Logcat window.
- Run the app and check the Logcat output.

Write a sum function

Write a function that will do the sum calculation from the previous section.

Do the following:

- Create a function named calcSum. It should not have any parameters or return type.
- Copy the sum related code from onCreate into the function.
- Call the function from onCreate.
- Run the app and check the Logcat output.

Use parameters and return type in the sum function

Make the following updates to the app:

- Add two parameters to the function from above. The first should be the start number and the second should be the end number.
- Add an Int return type to the function.
- Update the for loop so that it uses the start and end number parameters.
- Remove the println statement from the function.
- Return the sum at the end of the function.
- In onCreate, update the call to the function so that it passes in two parameters and uses the return type.
- Print the value returned by the function in the Logcat window.
- Run the app and check the Logcat output.

Create Employee class

Create an Employee class. Do the following:

- Right click the package in the Android view of the project on the left.
- Choose New | Kotlin Class/File.
- Type in Employee at the top of the dialog that appears, and press Enter. A new file with an empty Employee class will appear.
- Create member variables inside the class for id and name. Inside the class means inside the { }. Initialize id to 0 and name to "" at the declarations.
- Create an instance of Employee in onCreate and put data in it.

- Print the values of id and name in the Logcat window.
- Run the app and check the Logcat output.

Update Employee class

Update the Employee class as follows:

- Create a primary constructor that declares the member variables (instead of declaring them in the class body { }). The method body should now be empty since the variables are being declared as part of the primary constructor.
- Inside MainActivity.onCreate, update the code that creates the instances to use the primary constructor instead.
- Run the app and check the Logcat output.

Create a list of Employee

Do the following inside of MainActivity.onCreate:

- Create three instances of Employee.
- Declare a list of Employee objects. Use listOf to generate the list instance. Add two of the three instances of Employee to the list at the declaration (do not add all three).
- Print the list in the Logcat window. Use a for loop to iterate through the list.
- Run the app and check the Logcat output.
- Try and add the third employee list after the declaration. What happens? Is there an add method? Does autocomplete in Android Studio list add as a member choice?

Create a mutable list of Employee

Do the following inside of MainActivity.onCreate:

- Declare a mutable list of Employee objects. Make it empty to start.
- Create three instances of Employee.
- Add the three instances of Employee to the list.
- Print the list in the Logcat window. Use a for loop to iterate through the list.
- Run the app and check the Logcat output.

Create Department class

Create a Department class. Do the following:

- Create member variables for name(String) and emps(MutableList<Employee>). Declare them inside the { } (not in the primary constructor). You will be initializing them in an init block so do not set them to any values at the declaration. It will show an error but that is OK. The error will go away when you create the init block and give the variables values.

- Create an init block. Initialize both member variables in the init block. Initialize name to "empty" and initialize emps to an empty mutable list.
- Add method. This method should take an Employee as a parameter. It should add the given Employee to emps.
- Show method. In the Logcat window print the department name and then iterate through emps and display each Employee.
- In MainActivity.onCreate do the following:
    - Create an instance of Department.
    - Set the name on the Department instance.
    - Add employees to the Department instance (use the Department.add method to add the employees).
    - Call show to display the Department instance data.