UNIVERSITI MALAYSIA TERENGGANU

NATIVE MOBILE PROGRAMMING

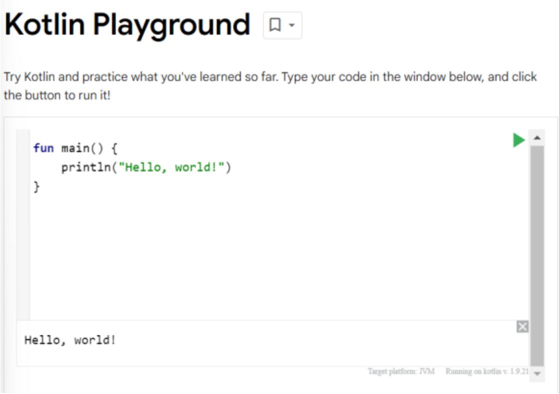
CSM 3123

NAME	: ILHAM HANINA MADIHA BINTI OTHMAN
MATRIC NO	: S63762
LECTURER	: DR RABIEI B MAMAT
LAB	: LAB 2

TASK 1: INTRODUCTION TO KOTLIN

Follow this link for the instructions. Introduction to programming in Kotlin | Android Basics Compose - First Android app | Android Developers

1. Run your first program:



Kotlin Playground

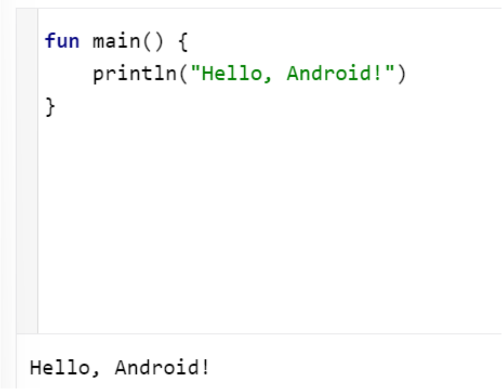
Try Kotlin and practice what you've learned so far. Type your code in the window below, and click the button to run it!

```
fun main() {
    println("Hello, world!")
}
```

```
Hello, world!
```

Target platform: JVM Running on kotlin v. 1.9.21

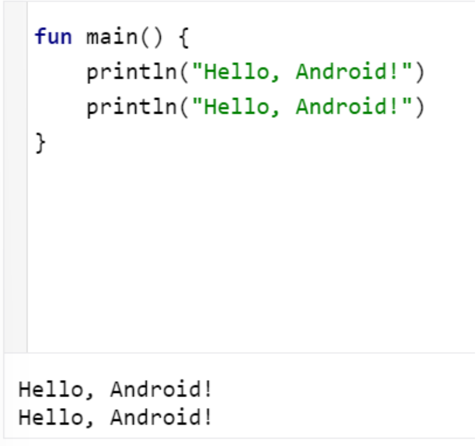
2. Parts of a function, try it to get Hello, Android!



```
fun main() {
    println("Hello, Android!")
}
```

```
Hello, Android!
```

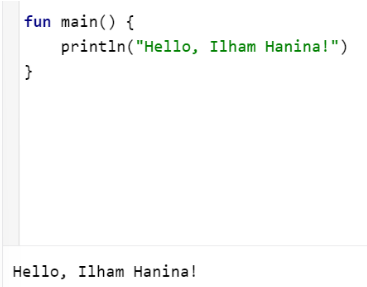
3. Print more than one messages.



```
fun main() {
    println("Hello, Android!")
    println("Hello, Android!")
}
```

```
Hello, Android!
Hello, Android!
```

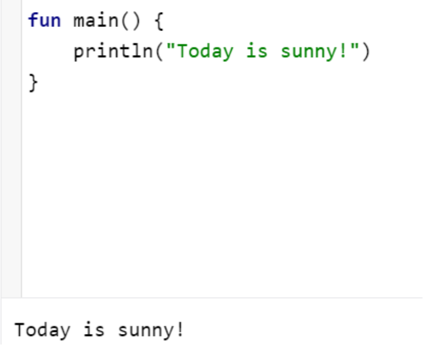
4. Change the code so that it says Hello, YOUR_NAME.



```
fun main() {
    println("Hello, Ilham Hanina!")
}
```

```
Hello, Ilham Hanina!
```

5. Fixing the code

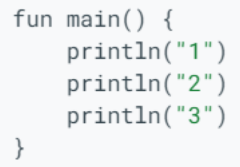


```
fun main() {
    println("Today is sunny!")
}
```

```
Today is sunny!
```

6. Exercises

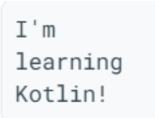
6.1 Read the code in this program and guess the output without running on Kotlin Playground.



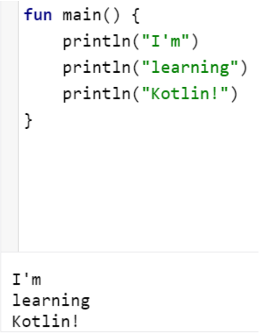
```
fun main() {
    println("1")
    println("2")
    println("3")
}
```

Output:
1
2
3

6.2 Use the Kotlin Playground to create a program that outputs the following messages.



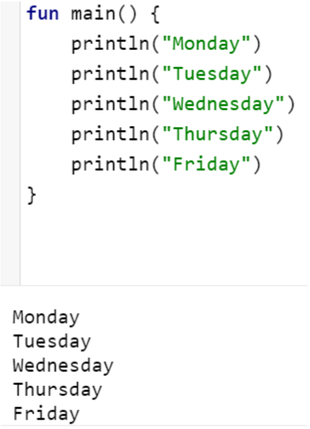
```
I'm
learning
Kotlin!
```



```
fun main() {
    println("I'm")
    println("learning")
    println("Kotlin!")
}
```

```
I'm
learning
Kotlin!
```

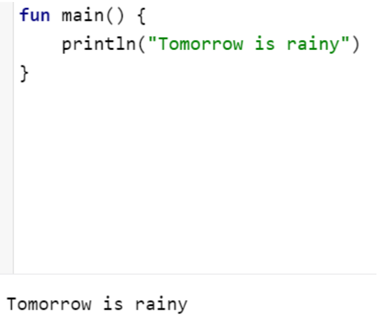
6.3 Fix the code so it will give the wanted output:



```
fun main() {
    println("Monday")
    println("Tuesday")
    println("Wednesday")
    println("Thursday")
    println("Friday")
}
```

```
Monday
Tuesday
Wednesday
Thursday
Friday
```

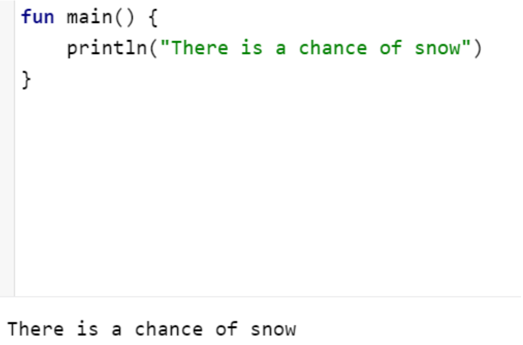
6.4 Fix the error so it gets the desired output:



```
fun main() {
    println("Tomorrow is rainy")
}
```

```
Tomorrow is rainy
```

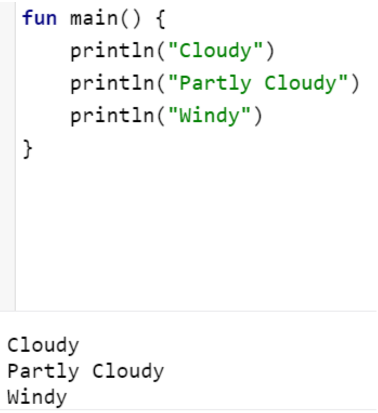
6.5 Fix the error of the program so it gets the desired output:



```
fun main() {
    println("There is a chance of snow")
}
```

```
There is a chance of snow
```

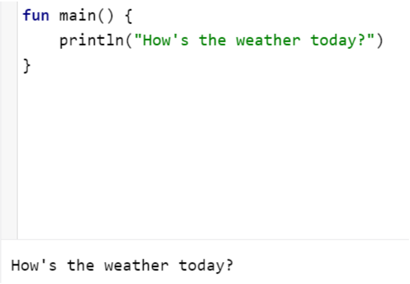
6.6 Fix the error of the program so it gets the desired output:



```
fun main() {
    println("Cloudy")
    println("Partly Cloudy")
    println("Windy")
}
```

```
Cloudy
Partly Cloudy
Windy
```

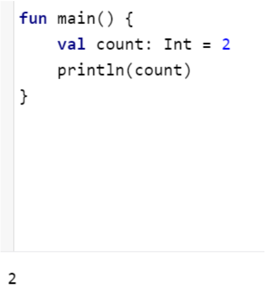
6.7 Fix the error of the program so it gets the desired output:



```
fun main() {
    println("How's the weather today?")
}
```

```
How's the weather today?
```

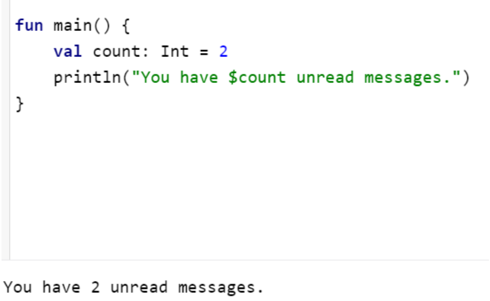
7. Define and use variables.



```
fun main() {
    val count: Int = 2
    println(count)
}
```

```
2
```

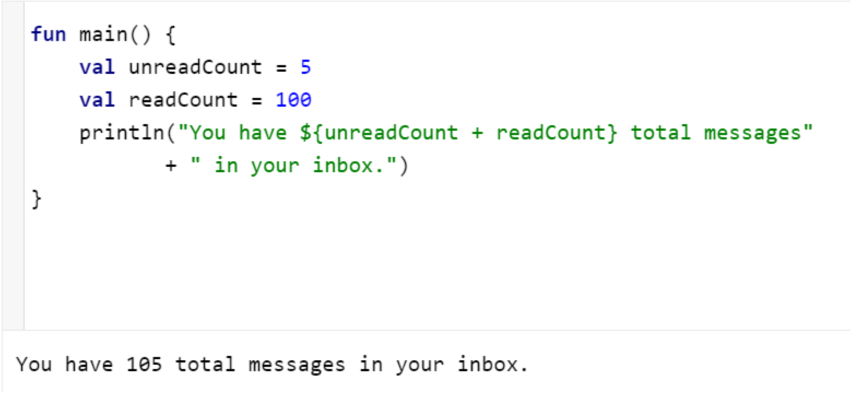
8. Use the dollar sign



```
fun main() {
    val count: Int = 2
    println("You have $count unread messages.")
}
```

```
You have 2 unread messages.
```

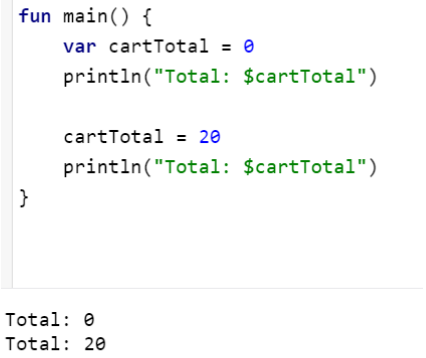
9. Basic math operations with integers



```
fun main() {
    val unreadCount = 5
    val readCount = 100
    println("You have ${unreadCount + readCount} total messages"
           + " in your inbox.")
}
```

```
You have 105 total messages in your inbox.
```

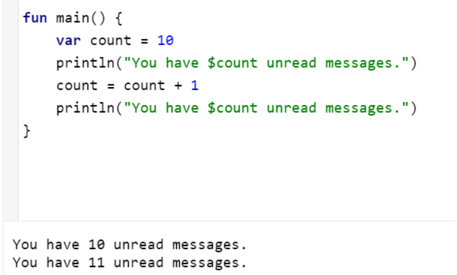
10. Var(cannot change the value) and val(the value is changeable)



```
fun main() {
    var cartTotal = 0
    println("Total: $cartTotal")

    cartTotal = 20
    println("Total: $cartTotal")
}
```

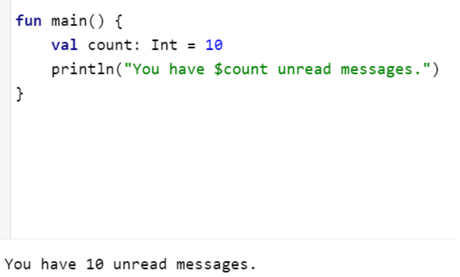
```
Total: 0
Total: 20
```

11. Increment and decrement



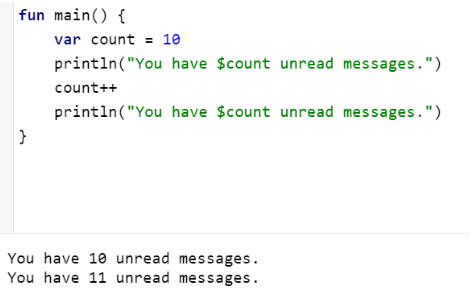
```
fun main() {
    var count = 10
    println("You have $count unread messages.")
    count = count + 1
    println("You have $count unread messages.")
}
```

```
You have 10 unread messages.
You have 11 unread messages.
```



```
fun main() {
    val count: Int = 10
    println("You have $count unread messages.")
}
```

```
You have 10 unread messages.
```

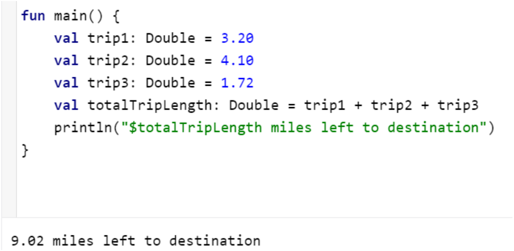


```
fun main() {
    var count = 10
    println("You have $count unread messages.")
    count++
    println("You have $count unread messages.")
}
```

```
You have 10 unread messages.
You have 11 unread messages.
```

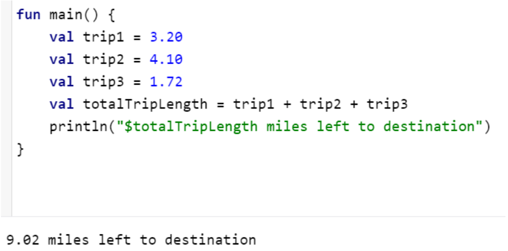
12. Explore other data types

Double:



```
fun main() {
    val trip1: Double = 3.20
    val trip2: Double = 4.10
    val trip3: Double = 1.72
    val totalTripLength: Double = trip1 + trip2 + trip3
    println("$totalTripLength miles left to destination")
}
```

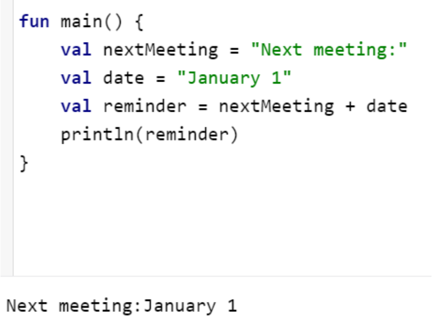
```
9.02 miles left to destination
```



```
fun main() {
    val trip1 = 3.20
    val trip2 = 4.10
    val trip3 = 1.72
    val totalTripLength = trip1 + trip2 + trip3
    println("$totalTripLength miles left to destination")
}
```

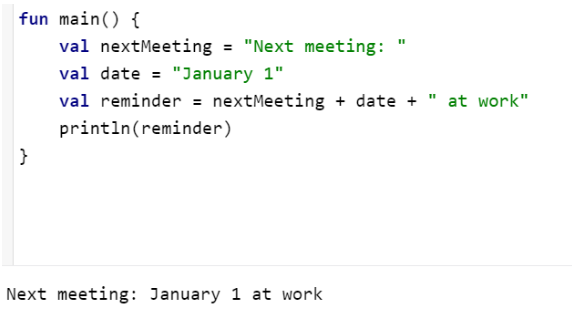
```
9.02 miles left to destination
```

String:



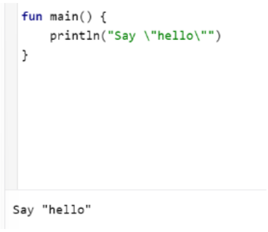
```
fun main() {
    val nextMeeting = "Next meeting:"
    val date = "January 1"
    val reminder = nextMeeting + date
    println(reminder)
}
```

```
Next meeting:January 1
```



```
fun main() {
    val nextMeeting = "Next meeting: "
    val date = "January 1"
    val reminder = nextMeeting + date + " at work"
    println(reminder)
}
```

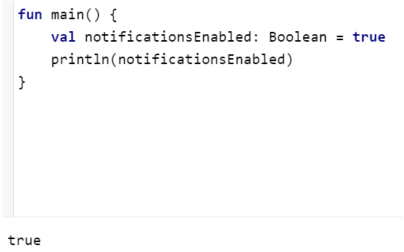
```
Next meeting: January 1 at work
```



```
fun main() {
    println("Say \"hello\"")
}
```

```
Say "hello"
```

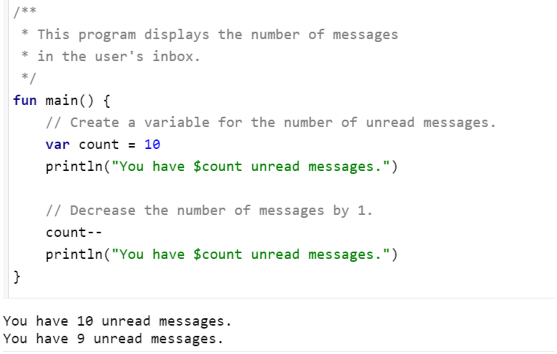
Boolean:



```
fun main() {
    val notificationsEnabled: Boolean = true
    println(notificationsEnabled)
}
```

```
true
```

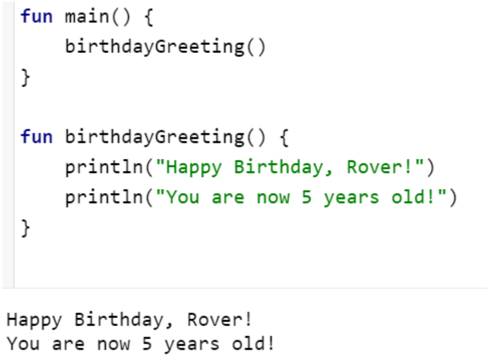
13. Commenting in your code



```
/**
 * This program displays the number of messages
 * in the user's inbox.
 */
fun main() {
    // Create a variable for the number of unread messages.
    var count = 10
    println("You have $count unread messages.")

    // Decrease the number of messages by 1.
    count--
    println("You have $count unread messages.")
}
```

```
You have 10 unread messages.
You have 9 unread messages.
```

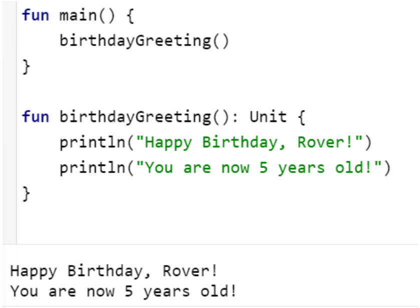
14. Define and call a function



```
fun main() {
    birthdayGreeting()
}

fun birthdayGreeting() {
    println("Happy Birthday, Rover!")
    println("You are now 5 years old!")
}
```

```
Happy Birthday, Rover!
You are now 5 years old!
```

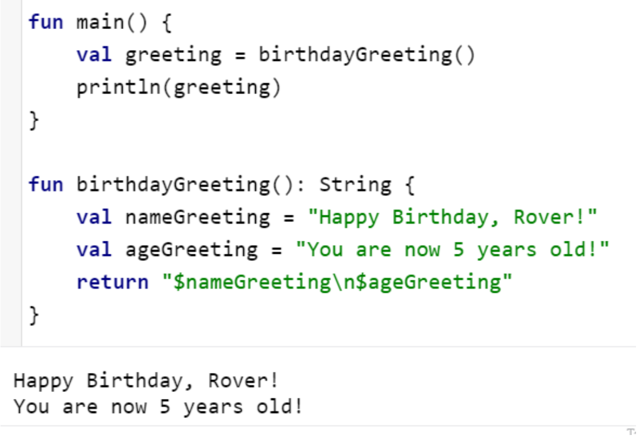
15. Using Unit (function that return a value)



```
fun main() {
    birthdayGreeting()
}

fun birthdayGreeting(): Unit {
    println("Happy Birthday, Rover!")
    println("You are now 5 years old!")
}
```

```
Happy Birthday, Rover!
You are now 5 years old!
```

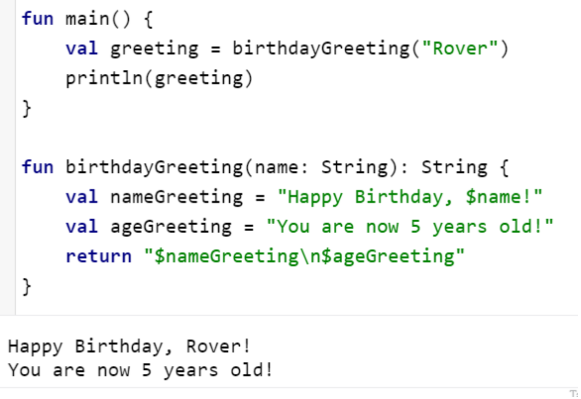
16. Using String, need to have return



```
fun main() {
    val greeting = birthdayGreeting()
    println(greeting)
}

fun birthdayGreeting(): String {
    val nameGreeting = "Happy Birthday, Rover!"
    val ageGreeting = "You are now 5 years old!"
    return "$nameGreeting\n$ageGreeting"
}
```

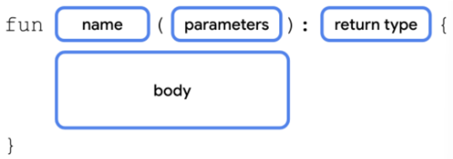
```
Happy Birthday, Rover!
You are now 5 years old!
```

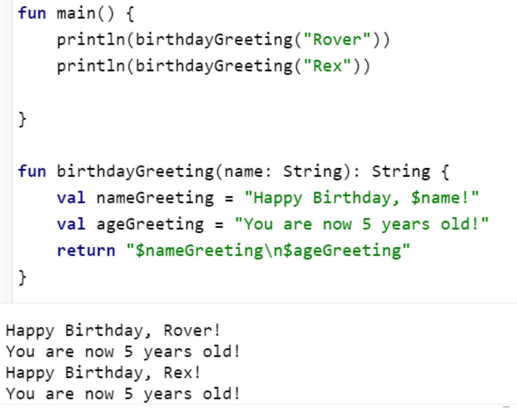
17. Adding parameter to the birthdayGreeting() function



```
fun main() {
    val greeting = birthdayGreeting("Rover")
    println(greeting)
}

fun birthdayGreeting(name: String): String {
    val nameGreeting = "Happy Birthday, $name!"
    val ageGreeting = "You are now 5 years old!"
    return "$nameGreeting\n$ageGreeting"
}
```

```
Happy Birthday, Rover!
You are now 5 years old!
```

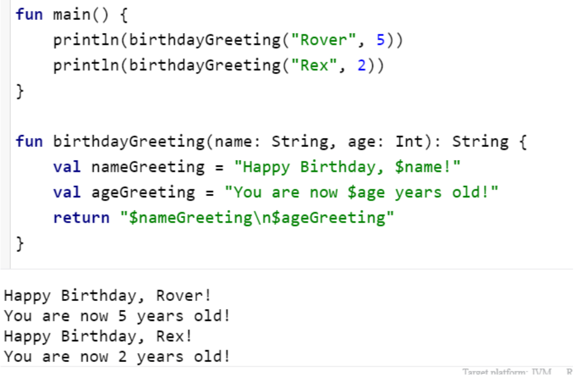




```
fun main() {
    println(birthdayGreeting("Rover"))
    println(birthdayGreeting("Rex"))

}

fun birthdayGreeting(name: String): String {
    val nameGreeting = "Happy Birthday, $name!"
    val ageGreeting = "You are now 5 years old!"
    return "$nameGreeting\n$ageGreeting"
}
```

```
Happy Birthday, Rover!
You are now 5 years old!
Happy Birthday, Rex!
You are now 5 years old!
```

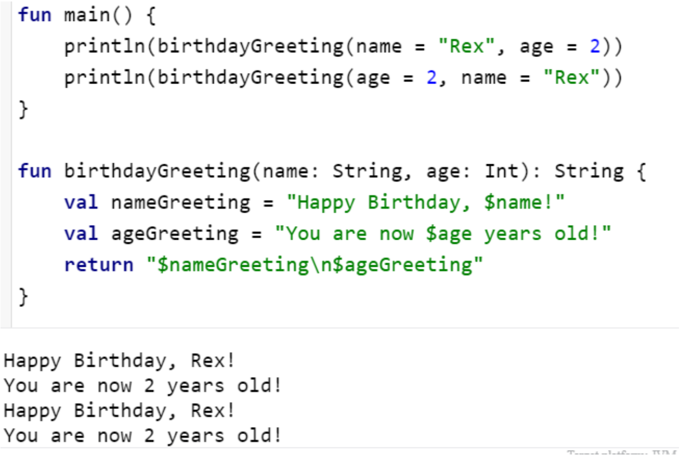
18. Functions with multiple parameters



```
fun main() {
    println(birthdayGreeting("Rover", 5))
    println(birthdayGreeting("Rex", 2))
}

fun birthdayGreeting(name: String, age: Int): String {
    val nameGreeting = "Happy Birthday, $name!"
    val ageGreeting = "You are now $age years old!"
    return "$nameGreeting\n$ageGreeting"
}
```

```
Happy Birthday, Rover!
You are now 5 years old!
Happy Birthday, Rex!
You are now 2 years old!
```

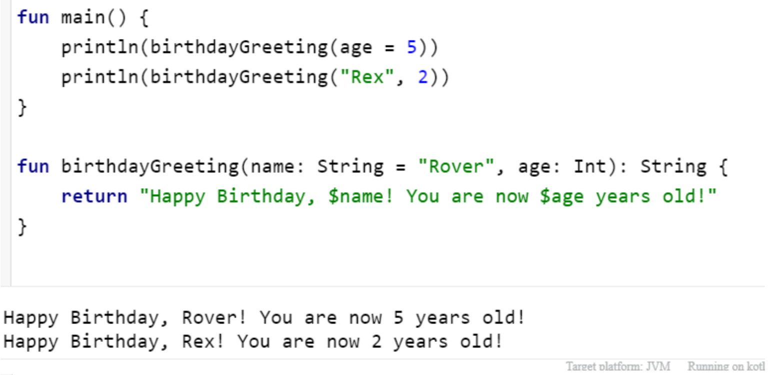
19. Named arguments



```
fun main() {
    println(birthdayGreeting(name = "Rex", age = 2))
    println(birthdayGreeting(age = 2, name = "Rex"))
}

fun birthdayGreeting(name: String, age: Int): String {
    val nameGreeting = "Happy Birthday, $name!"
    val ageGreeting = "You are now $age years old!"
    return "$nameGreeting\n$ageGreeting"
}
```

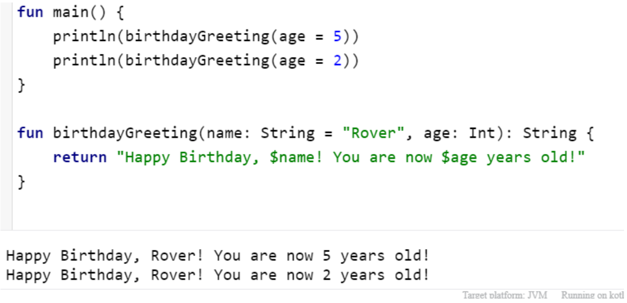
```
Happy Birthday, Rex!
You are now 2 years old!
Happy Birthday, Rex!
You are now 2 years old!
```

20. Default arguments



```
fun main() {
    println(birthdayGreeting(age = 5))
    println(birthdayGreeting("Rex", 2))
}

fun birthdayGreeting(name: String = "Rover", age: Int): String {
    return "Happy Birthday, $name! You are now $age years old!"
}
```

```
Happy Birthday, Rover! You are now 5 years old!
Happy Birthday, Rex! You are now 2 years old!
```

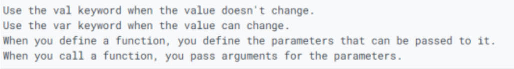


```
fun main() {
    println(birthdayGreeting(age = 5))
    println(birthdayGreeting(age = 2))
}

fun birthdayGreeting(name: String = "Rover", age: Int): String {
    return "Happy Birthday, $name! You are now $age years old!"
}
```

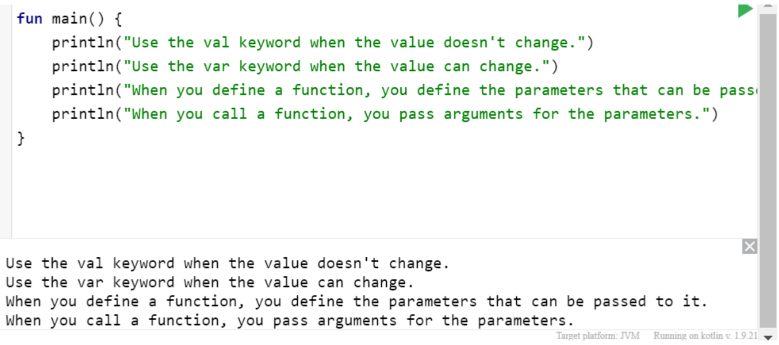
```
Happy Birthday, Rover! You are now 5 years old!
Happy Birthday, Rover! You are now 2 years old!
```

21. Practice Problems: Kotlin Basics

21.1 Can you write a main() function that prints these messages on four separate lines?



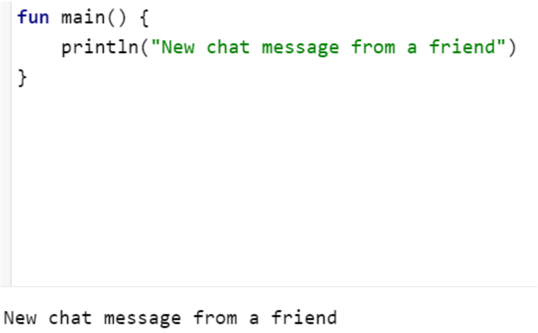
```
Use the val keyword when the value doesn't change.
Use the var keyword when the value can change.
When you define a function, you define the parameters that can be passed to it.
When you call a function, you pass arguments for the parameters.
```



```
fun main() {
    println("Use the val keyword when the value doesn't change.")
    println("Use the var keyword when the value can change.")
    println("When you define a function, you define the parameters that can be pass
    println("When you call a function, you pass arguments for the parameters.")
}
```

```
Use the val keyword when the value doesn't change.
Use the var keyword when the value can change.
When you define a function, you define the parameters that can be passed to it.
When you call a function, you pass arguments for the parameters.
```

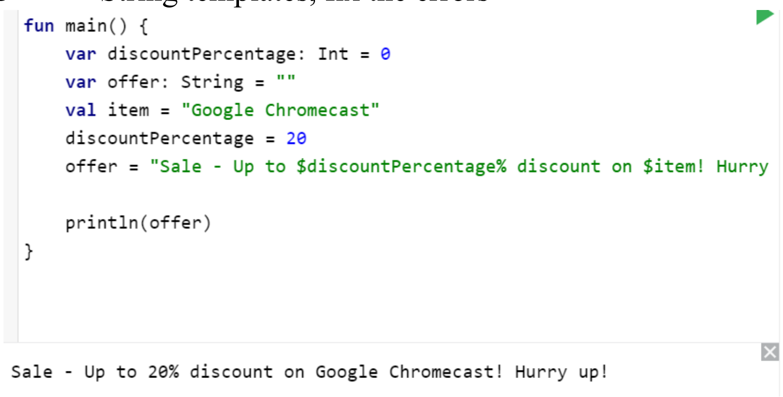
21.2 Fix compile error



```
fun main() {
    println("New chat message from a friend")
}
```

```
New chat message from a friend
```

21.3 String templates, fix the errors

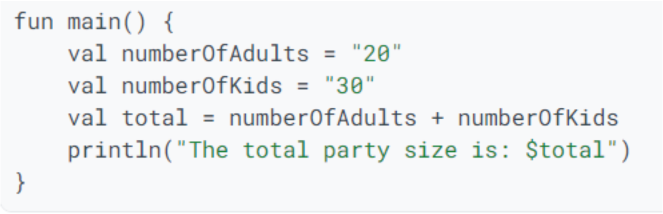


```
fun main() {
    var discountPercentage: Int = 0
    var offer: String = ""
    val item = "Google Chromecast"
    discountPercentage = 20
    offer = "Sale - Up to $discountPercentage% discount on $item! Hurry

    println(offer)
}
```

```
Sale - Up to 20% discount on Google Chromecast! Hurry up!
```

21.4 String concatenation

What is the output?

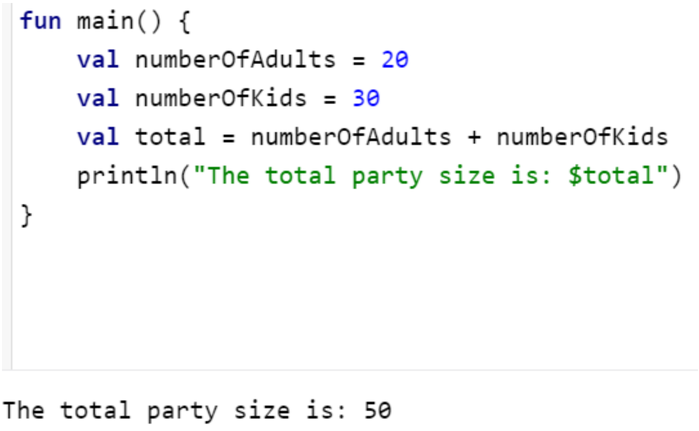


```
fun main() {
    val numberOfAdults = "20"
    val numberOfKids = "30"
    val total = numberOfAdults + numberOfKids
    println("The total party size is: $total")
}
```

Output:

The total party size is: 2030

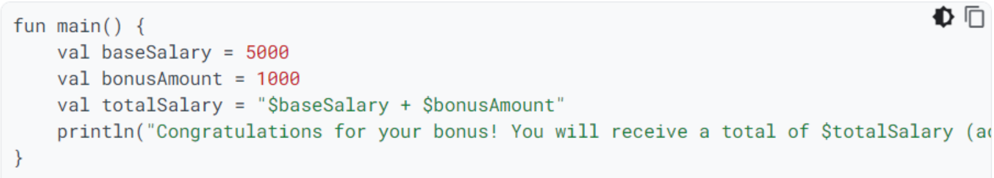
Fix the code



```
fun main() {
    val numberOfAdults = 20
    val numberOfKids = 30
    val total = numberOfAdults + numberOfKids
    println("The total party size is: $total")
}
```

```
The total party size is: 50
```

21.5 Message formatting

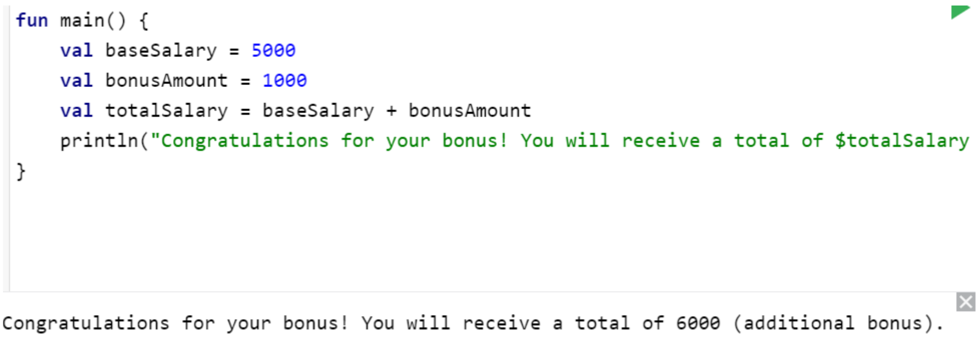


```
fun main() {
    val baseSalary = 5000
    val bonusAmount = 1000
    val totalSalary = "$baseSalary + $bonusAmount"
    println("Congratulations for your bonus! You will receive a total of $totalSalary (a
}
```

Output:

Congratulations for your bonus! You will receive a total of 5000 + 1000 (additional bonus).

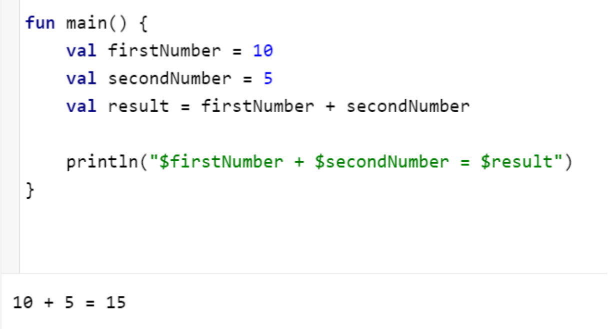
Fix the code:



```
fun main() {
    val baseSalary = 5000
    val bonusAmount = 1000
    val totalSalary = baseSalary + bonusAmount
    println("Congratulations for your bonus! You will receive a total of $totalSalary
}
```

```
Congratulations for your bonus! You will receive a total of 6000 (additional bonus).
```

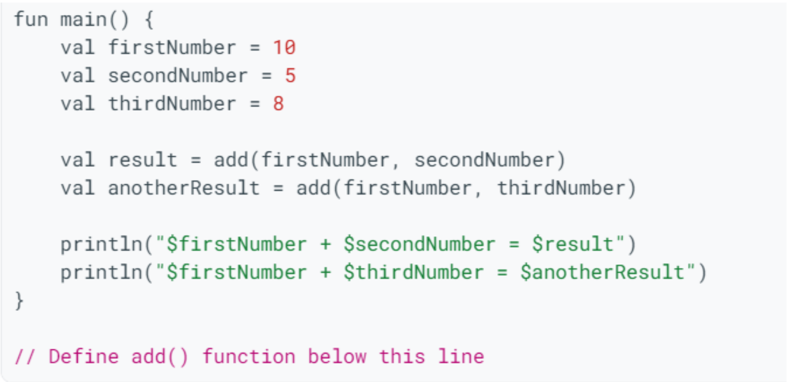
21.6 Implement basic math operations



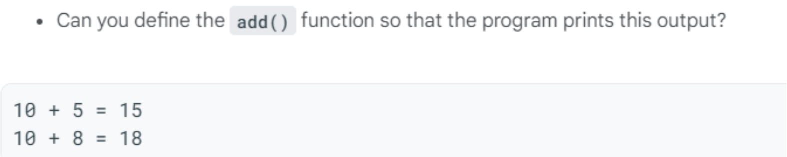
```
fun main() {
    val firstNumber = 10
    val secondNumber = 5
    val result = firstNumber + secondNumber

    println("$firstNumber + $secondNumber = $result")
}
```

```
10 + 5 = 15
```

Add add() function

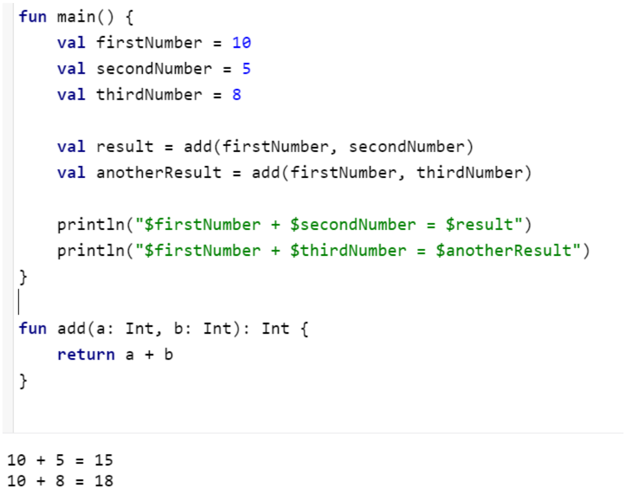


```
fun main() {
    val firstNumber = 10
    val secondNumber = 5
    val thirdNumber = 8

    val result = add(firstNumber, secondNumber)
    val anotherResult = add(firstNumber, thirdNumber)

    println("$firstNumber + $secondNumber = $result")
    println("$firstNumber + $thirdNumber = $anotherResult")
}

// Define add() function below this line
```



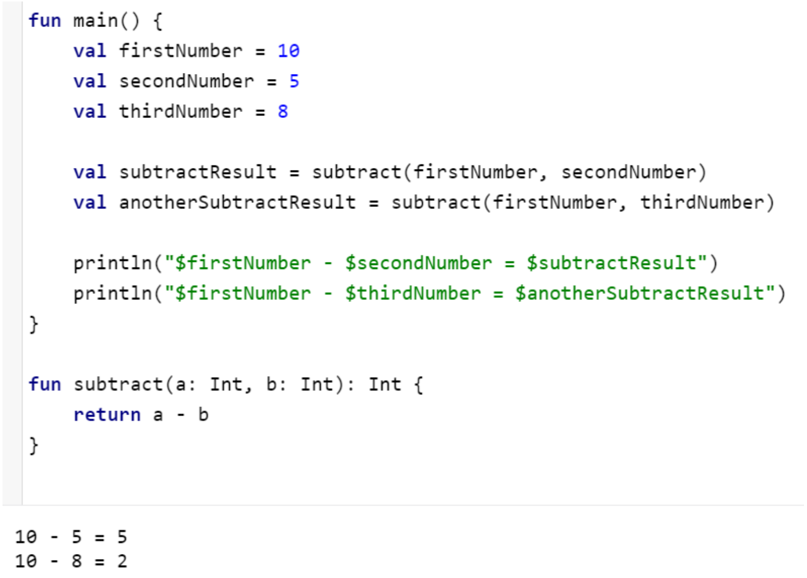
- Can you define the `add()` function so that the program prints this output?

```
10 + 5 = 15
10 + 8 = 18
```



```
fun main() {
    val firstNumber = 10
    val secondNumber = 5
    val thirdNumber = 8

    val result = add(firstNumber, secondNumber)
    val anotherResult = add(firstNumber, thirdNumber)

    println("$firstNumber + $secondNumber = $result")
    println("$firstNumber + $thirdNumber = $anotherResult")
}

fun add(a: Int, b: Int): Int {
    return a + b
}
```

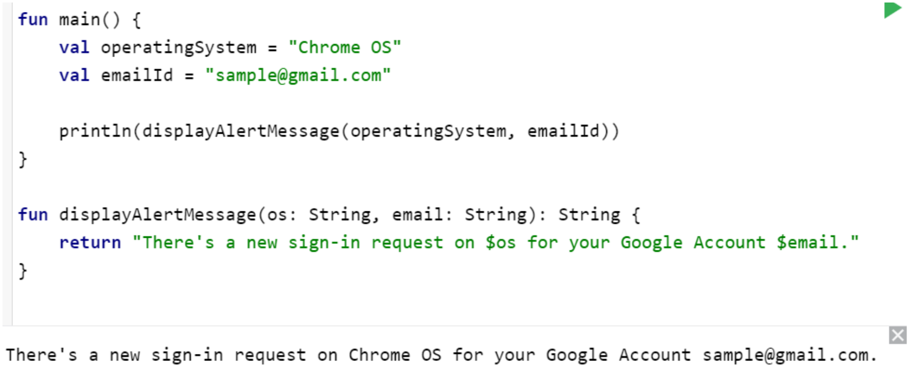
```
10 + 5 = 15
10 + 8 = 18
```



```
fun main() {
    val firstNumber = 10
    val secondNumber = 5
    val thirdNumber = 8

    val subtractResult = subtract(firstNumber, secondNumber)
    val anotherSubtractResult = subtract(firstNumber, thirdNumber)

    println("$firstNumber - $secondNumber = $subtractResult")
    println("$firstNumber - $thirdNumber = $anotherSubtractResult")
}

fun subtract(a: Int, b: Int): Int {
    return a - b
}
```

```
10 - 5 = 5
10 - 8 = 2
```

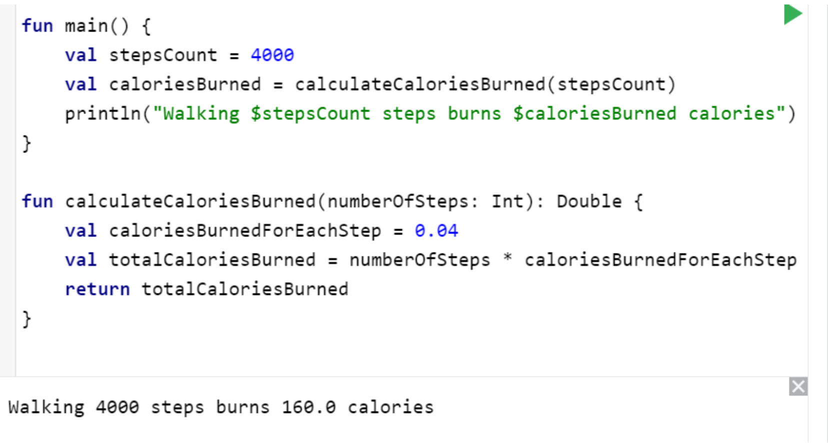
21.7 Default parameters



```
fun main() {
    val operatingSystem = "Chrome OS"
    val emailId = "sample@gmail.com"

    println(displayAlertMessage(operatingSystem, emailId))
}

fun displayAlertMessage(os: String, email: String): String {
    return "There's a new sign-in request on $os for your Google Account $email."
}
```

```
There's a new sign-in request on Chrome OS for your Google Account sample@gmail.com.
```

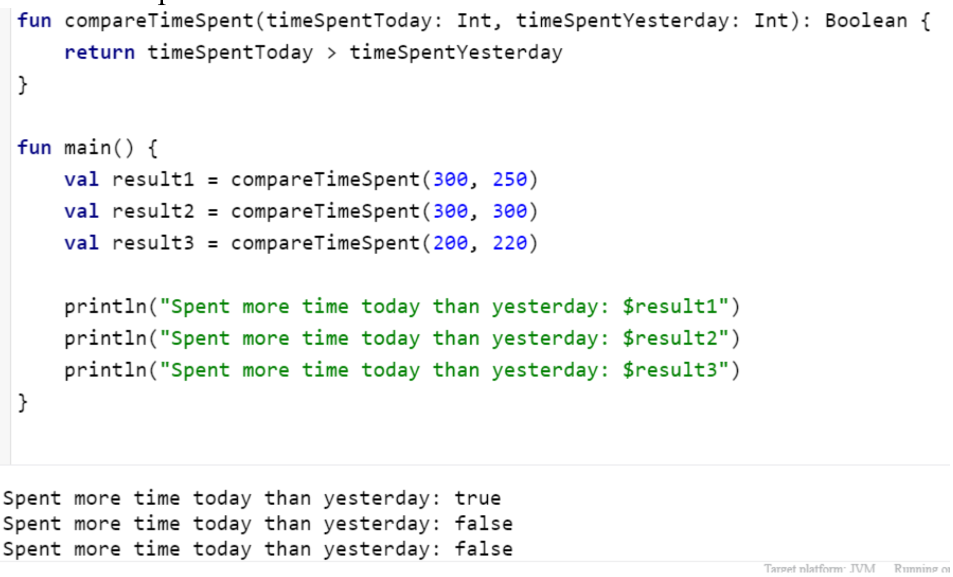
21.8 Pedometer



```kotlin
fun main() {
    val stepsCount = 4000
    val caloriesBurned = calculateCaloriesBurned(stepsCount)
    println("Walking $stepsCount steps burns $caloriesBurned calories")
}

fun calculateCaloriesBurned(numberOfSteps: Int): Double {
    val caloriesBurnedForEachStep = 0.04
    val totalCaloriesBurned = numberOfSteps * caloriesBurnedForEachStep
    return totalCaloriesBurned
}
```

```
Walking 4000 steps burns 160.0 calories
```

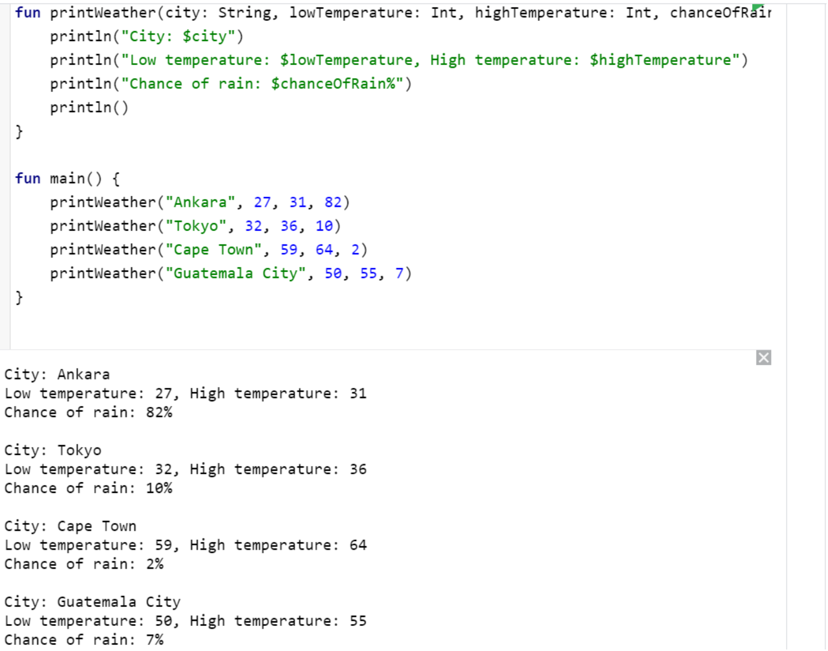
21.9 Compare two numbers



```kotlin
fun compareTimeSpent(timeSpentToday: Int, timeSpentYesterday: Int): Boolean {
    return timeSpentToday > timeSpentYesterday
}

fun main() {
    val result1 = compareTimeSpent(300, 250)
    val result2 = compareTimeSpent(300, 300)
    val result3 = compareTimeSpent(200, 220)

    println("Spent more time today than yesterday: $result1")
    println("Spent more time today than yesterday: $result2")
    println("Spent more time today than yesterday: $result3")
}
```

```
Spent more time today than yesterday: true
Spent more time today than yesterday: false
Spent more time today than yesterday: false
```

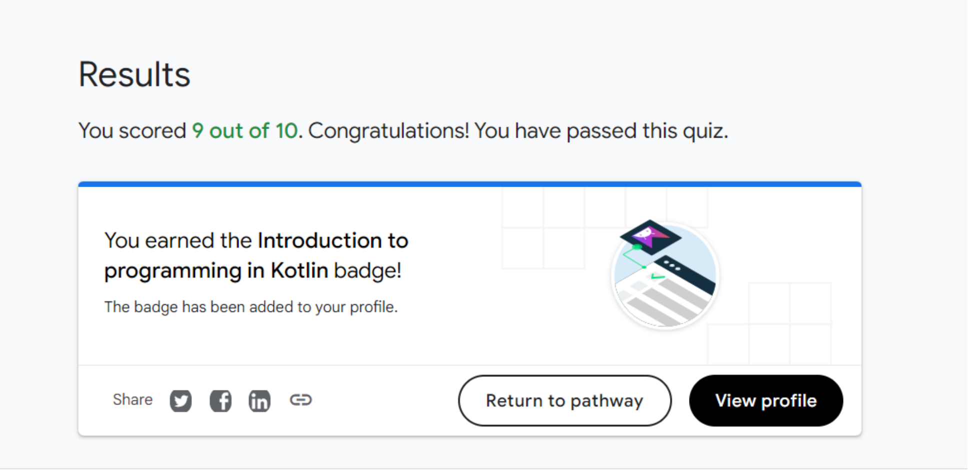
21.10 Move duplicate code into a function



```kotlin
fun printWeather(city: String, lowTemperature: Int, highTemperature: Int, chanceOfRai
    println("City: $city")
    println("Low temperature: $lowTemperature, High temperature: $highTemperature")
    println("Chance of rain: $chanceOfRain%")
    println()
}

fun main() {
    printWeather("Ankara", 27, 31, 82)
    printWeather("Tokyo", 32, 36, 10)
    printWeather("Cape Town", 59, 64, 2)
    printWeather("Guatemala City", 50, 55, 7)
}
```

```
City: Ankara
Low temperature: 27, High temperature: 31
Chance of rain: 82%

City: Tokyo
Low temperature: 32, High temperature: 36
Chance of rain: 10%

City: Cape Town
Low temperature: 59, High temperature: 64
Chance of rain: 2%

City: Guatemala City
Low temperature: 50, High temperature: 55
Chance of rain: 7%
```

22. Screenshot of the result for quiz for Introduction to Programming in Kotlin



Results

You scored 9 out of 10. Congratulations! You have passed this quiz.

You earned the **Introduction to programming in Kotlin** badge!

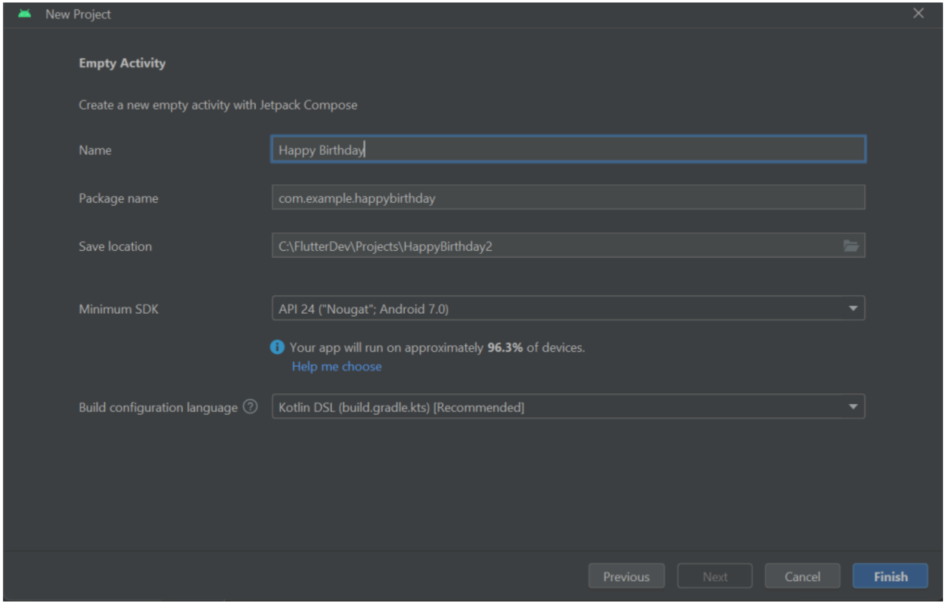
The badge has been added to your profile.

Was this helpful?

TASK 2: USING BASIC LAYOUT

Follow this link for the instructions. [Build a basic layout | Android Basics Compose - First Android app | Android Developers]()

1. Build a simple app with text composables
   Create an Empty Activity Project named Happy Birthday



Empty Activity

Create a new empty activity with Jetpack Compose

Name: Happy Birthday

Package name: com.example.happybirthday

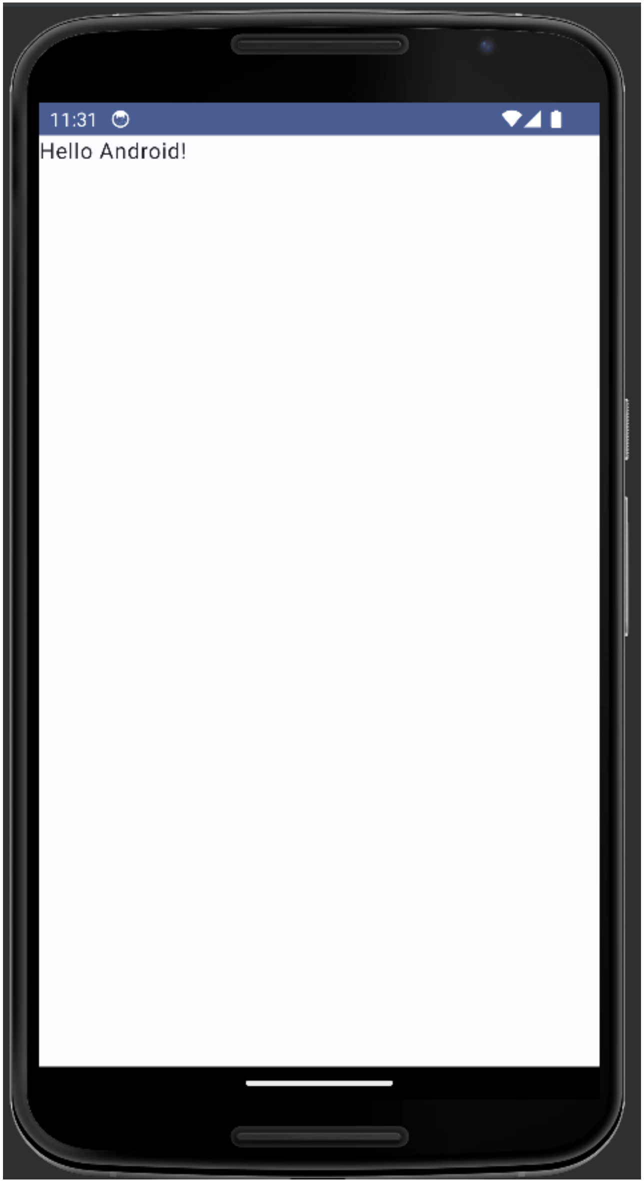
Save location: C:\FlutterDev\Projects\HappyBirthday2

Minimum SDK: API 24 ("Nougat"; Android 7.0)

Your app will run on approximately **96.3%** of devices.
Help me choose

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

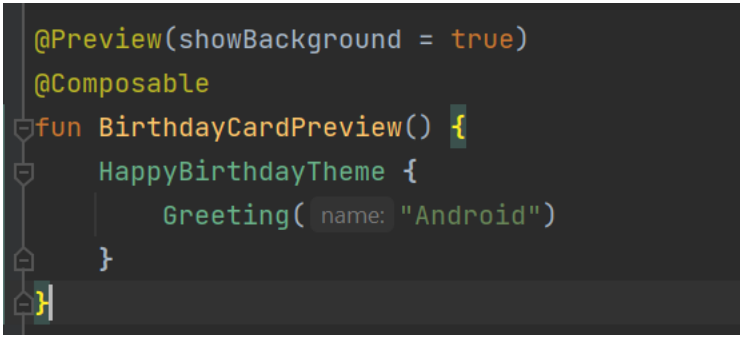
Run the project and we will get this on our emulator:



11:31

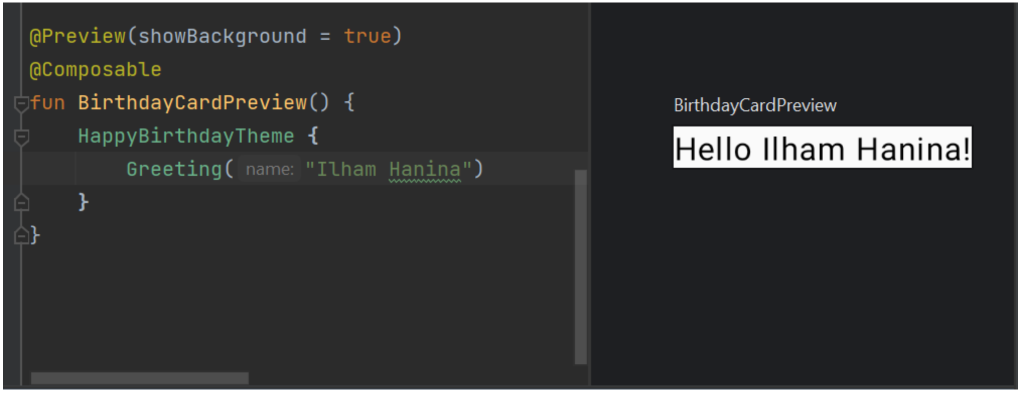
Hello Android!

In the MainActivity.kt, change the GreetingPreview() function and change the name to BirthdayCardPreview() since it is good practice for functions to be name by their functionality.



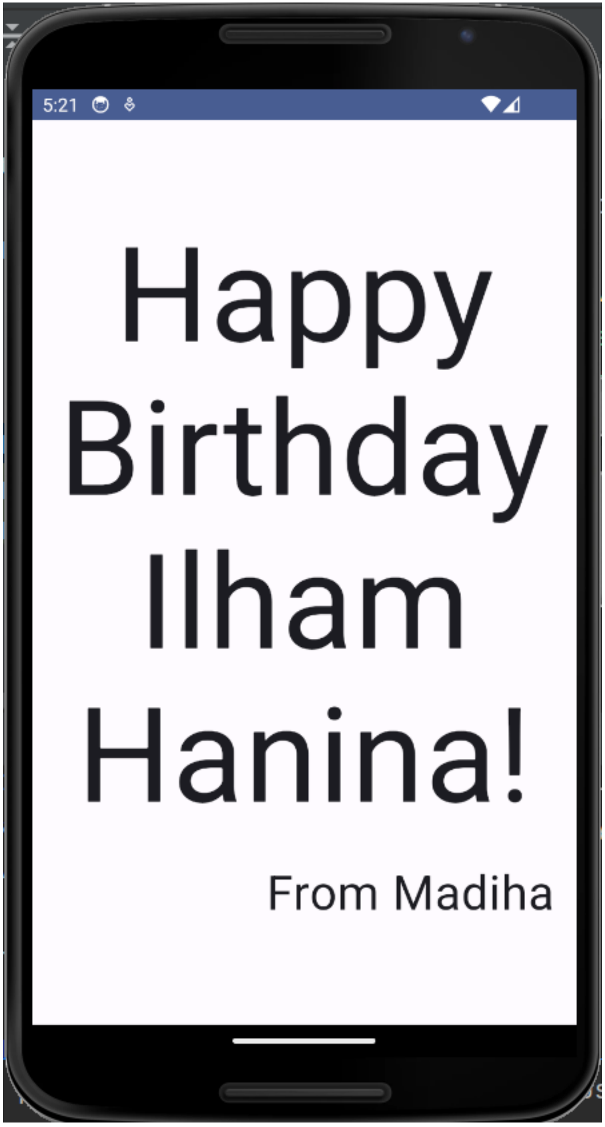
```
@Preview(showBackground = true)
@Composable
fun BirthdayCardPreview() {
    HappyBirthdayTheme {
        Greeting( name: "Android")
    }
}
```

Change the “Android” in the Greeting() function with your name.



```
@Preview(showBackground = true)
@Composable
fun BirthdayCardPreview() {
    HappyBirthdayTheme {
        Greeting( name: "Ilham Hanina")
    }
}
```

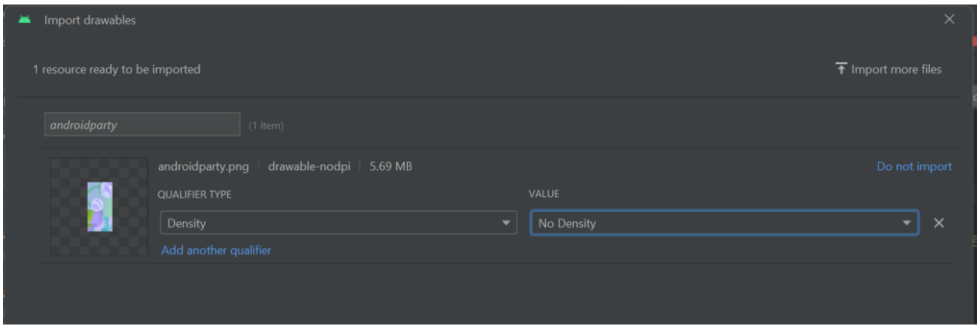
After changing the code based on the guide, this is the output:



Happy Birthday Ilham Hanina!

From Madiha

2. Add images to your Android app



Import drawables

1 resource ready to be imported

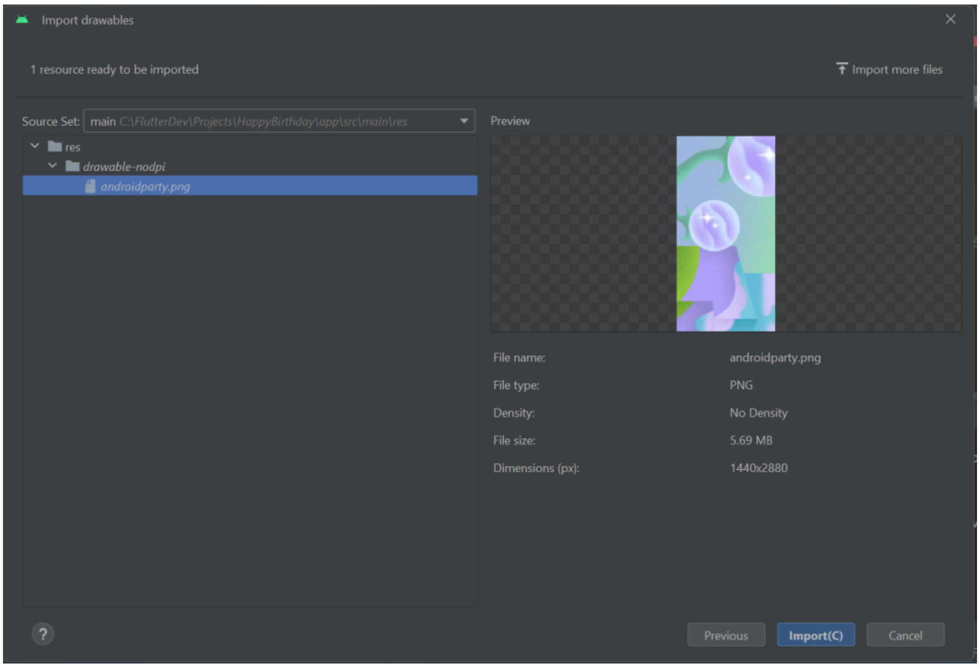
androidparty (1 item)

androidparty.png | drawable-nodpi | 5.69 MB

QUALIFIER TYPE: Density

VALUE: No Density

Click Import,



Import drawables

1 resource ready to be imported

Source Set: main C:\FlutterDev\Projects\HappyBirthday\app\src\main\res

```
res
  drawable-nodpi
    androidparty.png
```

Preview

File name: androidparty.png

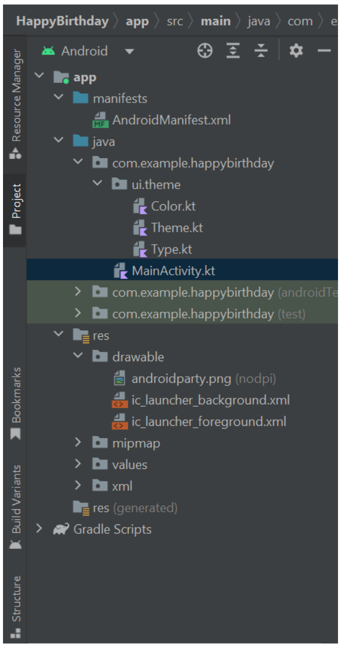
File type: PNG

Density: No Density

File size: 5.69 MB

Dimensions (px): 1440x2880

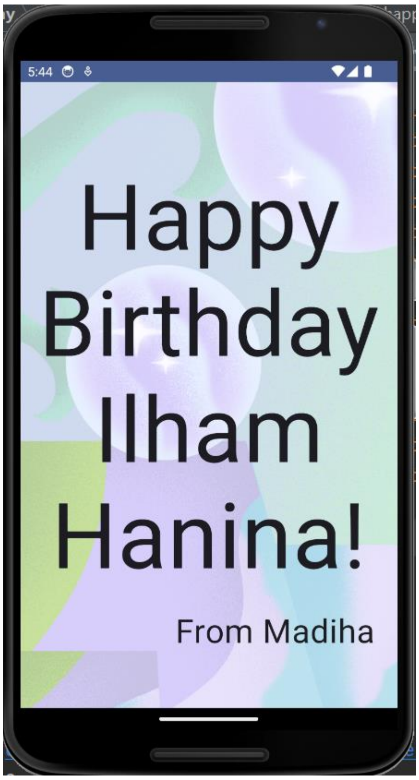
This is how the project directory should be for the imported image:



HappyBirthday 〉 app 〉 src 〉 main 〉 java 〉 com 〉 e

```
app
  manifests
    AndroidManifest.xml
  java
    com.example.happybirthday
      ui.theme
        Color.kt
        Theme.kt
        Type.kt
      MainActivity.kt
    com.example.happybirthday (androidTe
    com.example.happybirthday (test)
  res
    drawable
      androidparty.png (nodpi)
      ic_launcher_background.xml
      ic_launcher_foreground.xml
    mipmap
    values
    xml
  res (generated)
Gradle Scripts
```

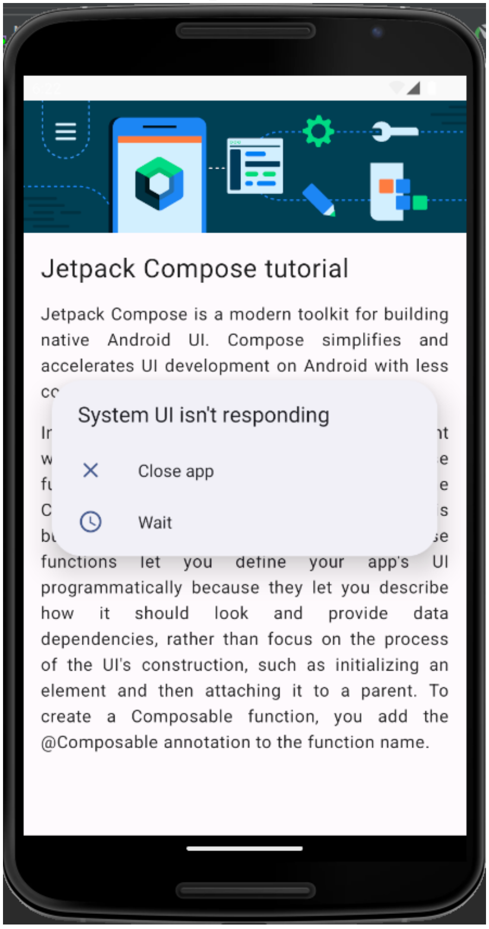
The output:



Happy Birthday Ilham Hanina!

From Madiha

3. Practice: Compose Basic
   Practice 1: Compose Article



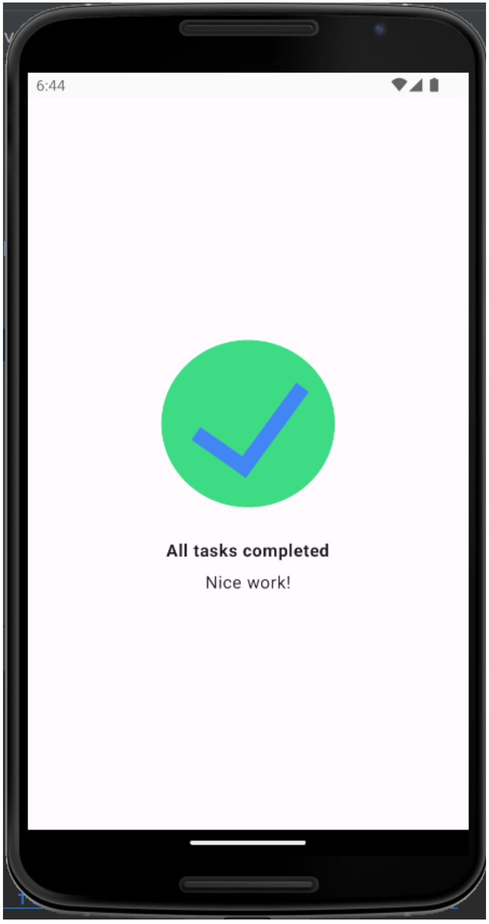
Jetpack Compose tutorial

Jetpack Compose is a modern toolkit for building native Android UI. Compose simplifies and accelerates UI development on Android with less co

Ir nt
w e
fu e
C s
bu se
functions let you define your app's UI programmatically because they let you describe how it should look and provide data dependencies, rather than focus on the process of the UI's construction, such as initializing an element and then attaching it to a parent. To create a Composable function, you add the @Composable annotation to the function name.

System UI isn't responding

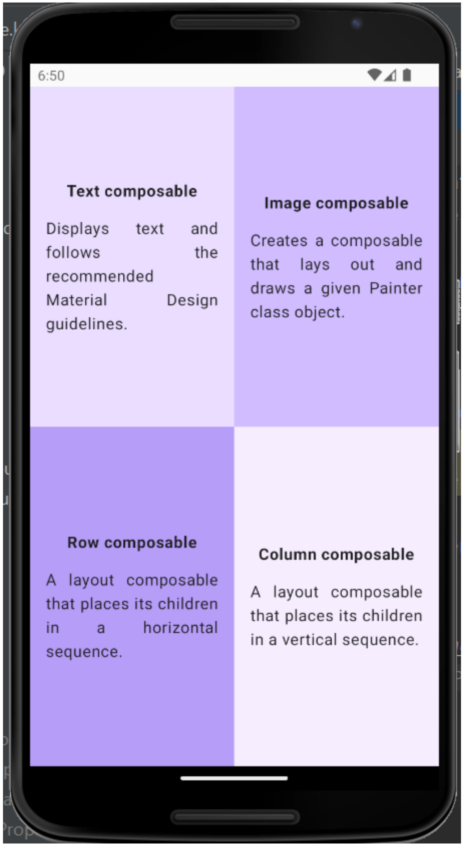
Practice 2: Task Manager



All tasks completed

Nice work!

Practice 3: Compose quadrant



Text composable

Displays text and follows the recommended Material Design guidelines.

Image composable

Creates a composable that lays out and draws a given Painter class object.

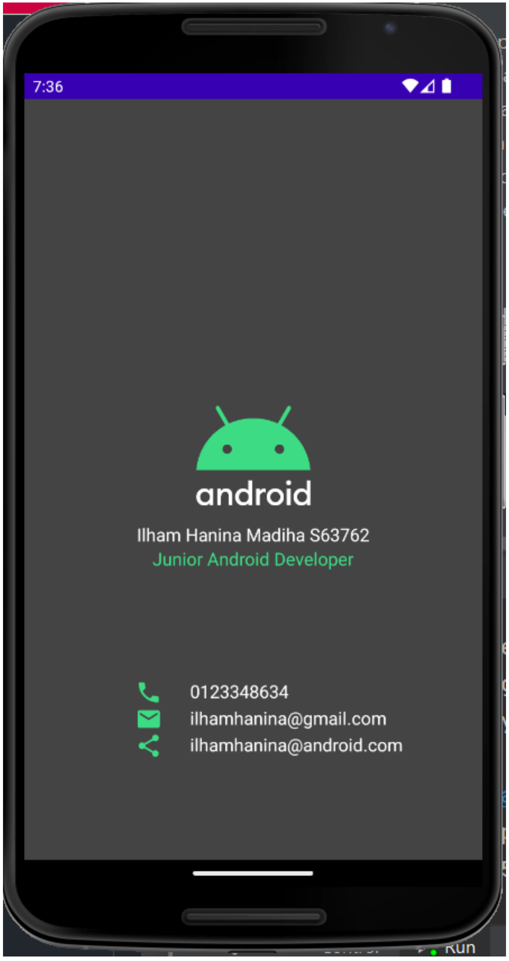
Row composable

A layout composable that places its children in a horizontal sequence.

Column composable

A layout composable that places its children in a vertical sequence.

4. Project: Create a business card app



7:36

android

Ilham Hanina Madiha S63762

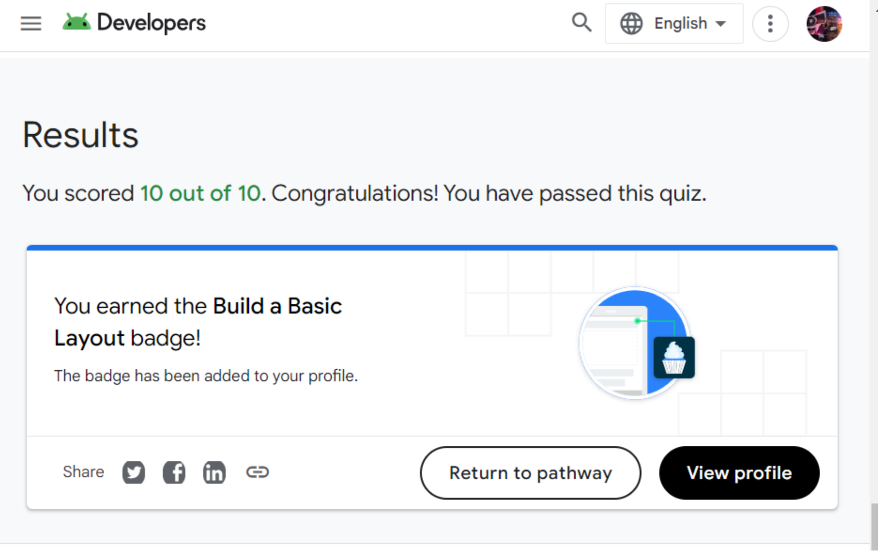
Junior Android Developer

☎ 0123348634

ilhamhanina@gmail.com

ilhamhanina@android.com

5. Result for quiz



Results

You scored 10 out of 10. Congratulations! You have passed this quiz.

You earned the Build a Basic Layout badge!

The badge has been added to your profile.

Submission

Github Link:

https://github.com/ilhamhanina/CSM3123NativeMobileProgramming.git