



UNIVERSITI MALAYSIA TERENGGANU

FACULTY OF OCEAN ENGINEERING TECHNOLOGY & INFORMATICS

SEMESTER 1 2023/2024

NATIVE MOBILE PROGRAMMING

CSM 3123

LAB 3 REPORT

PREPARED FOR:

DR. RABIEI BIN MAMAT

PREPARED BY:

MOHAMAD HAZIM BIN MOHD SHAKRI (S61770)

3. WRITE CONDITIONALS IN KOTLIN

(USE IF/ELSE STATEMENTS TO EXPRESS CONDITIONS)

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
/**  
 * You can edit, run, and share this code.  
 * play.kotlinlang.org  
 */  
fun main() {  
    println(1 == 1)  
}
```

The output window below the editor shows the result of the execution: "true". The operating system taskbar at the bottom indicates it's running on a Windows 10 machine with a weather forecast of 26°C Partly cloudy.

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
/**  
 * You can edit, run, and share this code.  
 * play.kotlinlang.org  
 */  
fun main() {  
    println(1 < 1)  
}
```

The output window below the editor shows the result of the execution: "false". The operating system taskbar at the bottom indicates it's running on a Windows 10 machine with a weather forecast of 26°C Partly cloudy.

The screenshot shows the Kotlin Playground interface. At the top, there's a navigation bar with links for Solutions, Docs, Community, Teach, Play, and a search icon. Below the navigation bar, the version is set to 1.9.10 and the JVM is selected. A "Program arguments" input field contains the URL play.kotlinlang.org. On the right, there are buttons for Copy link, Share code, and Run. The main area displays the following Kotlin code:

```
* You can edit, run, and share this code.
* play.kotlinlang.org
*/
fun main() {
    val trafficColor = "Red"
    if (trafficColor == "Red") {
        println("Stop")
    }
}
```

Below the code, a terminal window shows the output "Stop". The status bar at the bottom indicates it's 10:15 AM on 10/11/2023.

This screenshot shows the same setup as the first one, but the code has been modified to include an else block:

```
/*
fun main() {
    val trafficColor = "Red"
    if (trafficColor == "Red") {
        println("Stop")
    } else {
        println("Go")
    }
}
```

The terminal window still shows "Stop" as the output. The status bar at the bottom indicates it's 10:16 AM on 10/11/2023.

A screenshot of the Kotlin Playground interface. The code editor shows the following Kotlin code:

```
/*
fun main() {
    val trafficColor = "Green"
    if (trafficColor == "Red") {
        println("Stop")
    } else {
        println("Go")
    }
}
```

The code is run, and the output window below shows the word "Go". The status bar at the bottom indicates it's 10:17 AM on October 11, 2023, with a weather of 26°C and partly cloudy.

A screenshot of the Kotlin Playground interface. The code editor shows the following Kotlin code:

```
fun main() {
    val trafficLightColor = "Green"
    if (trafficLightColor == "Red") {
        println("Stop")
    } else if (trafficLightColor == "Yellow") {
        println("Caution")
    } else {
        println("Go")
    }
}
```

The code is run, and the output window below shows the word "Go". The status bar at the bottom indicates it's 10:20 AM on October 11, 2023, with a weather of 26°C and partly cloudy.

Kotlin Playground: Edit, Run, Share

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    val trafficLightColor = "Green"
    if (trafficLightColor == "Red") {
        println("Stop")
    } else if (trafficLightColor == "Yellow") {
        println("Slow")
    } else {
        println("Go")
    }
}
```

Go

26°C Partly cloudy Q Search ENG US 10/11/2023

Kotlin Playground: Edit, Run, Share

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    val trafficLightColor = "Yellow"
    if (trafficLightColor == "Red") {
        println("Stop")
    } else if (trafficLightColor == "Yellow") {
        println("Slow")
    } else {
        println("Go")
    }
}
```

Slow

26°C Partly cloudy Q Search ENG US 10/11/2023

Kotlin Playground: Edit, Run, Share

play.kotlinlang.org/eyJ2ZXJzaW9uIjoiMS45LjEwliwicGxhdGZvcmUiOiJqYXZhliwiYXJnci6lilsIm5vbmcVNXYJrZXJzIjp0cnVILCJ0aGVtZSI6ImlkZWEl...

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    val trafficLightColor = "Black"
    if (trafficLightColor == "Red") {
        println("Stop")
    } else if (trafficLightColor == "Yellow") {
        println("Slow")
    } else {
        println("Go")
    }
}
```

Go

Weather alert In effect

Q Search Ps AI 10/11/2023 10:23 AM ENG US

Kotlin Playground: Edit, Run, Share

play.kotlinlang.org/eyJ2ZXJzaW9uIjoiMS45LjEwliwicGxhdGZvcmUiOiJqYXZhliwiYXJnci6lilsIm5vbmcVNXYJrZXJzIjp0cnVILCJ0aGVtZSI6ImlkZWEl...

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
/*
 * You can edit, run, and share this code.
 * play.kotlinlang.org
 */
fun main() {
    val trafficLightColor = "Black"
    if (trafficLightColor == "Red") {
        println("Stop")
    } else if (trafficLightColor == "Yellow") {
        println("Slow")
    } else if (trafficLightColor == "Green") {
        println("Go")
    }
}
```

26°C Partly cloudy

Q Search Ps AI 10/11/2023 10:24 AM ENG US

Kotlin Playground: Edit, Run, Share

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    val trafficLightColor = "Black"
    if (trafficLightColor == "Red") {
        println("Stop")
    } else if (trafficLightColor == "Yellow") {
        println("Slow")
    } else if (trafficLightColor == "Green") {
        println("Go")
    } else {
        println("Invalid traffic-Light color")
}
```

Invalid traffic-Light color

26°C Near record Q Search ENG US 10/11/2023

Kotlin Playground: Edit, Run, Share

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    val trafficLightColor = "Black"
    if (trafficLightColor == "Blue") {
        println("Stop")
    } else if (trafficLightColor == "Magenta") {
        println("Slow")
    } else if (trafficLightColor == "Green") {
        println("Go")
    } else {
        println("Invalid traffic-Light color")
}
```

Invalid traffic-Light color

26°C Partly cloudy Q Search ENG US 10/11/2023

USE A WHEN STATEMENT FOR MULTIPLE BRANCHES

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
1.9.10 JVM Program arguments
*/  
fun main() {  
    val trafficLightColor = "Yellow"  
  
    when (trafficLightColor) {  
        "Red" -> println("Stop")  
        "Yellow" -> println("Slow")  
        "Green" -> println("Go")  
        else -> println ("Invalid traffic-Light color")  
    }  
}
```

The output window below the editor shows the result of running the program: "Slow". The status bar at the bottom indicates it's 26°C and Partly cloudy.

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
1.9.10 JVM Program arguments
Course: PENGATURCARAAN MUDA  
play.kotlinlang.org/#eyJ2ZXJzaW9uljoiMS45LjEwliwicGxhdGZvcm0iOjqYXZhliwYXJncy6iilsIm5vb...  
+  
Kotlin Solutions Docs Community Teach Play  
Copy link Share code Run  
  
fun main() {  
    val x = 3  
    when (x){  
        2 -> println("x is a prime number between 1 and 10.")  
        3 -> println("x is a prime number between 1 and 10.")  
        5 -> println("x is a prime number between 1 and 10.")  
        7 -> println("x is a prime number between 1 and 10.")  
        else -> println("x isn't a prime number between 1 and 10.")  
    }  
}  
  
x is a prime number between 1 and 10.
```

A cookie consent dialog is visible on the right side of the screen, containing text about data collection and cookies, and buttons for "[A]ccept All" and "[M]anage Settings". The status bar at the bottom indicates it's 26°C and Partly cloudy.

The screenshot shows the Kotlin Playground interface. The top navigation bar includes links for Solutions, Docs, Community, Teach, Play, and a search bar. Below the navigation is a toolbar with version (1.9.10), JVM dropdown, and a "Program arguments" input field. To the right are "Copy link", "Share code", and a prominent purple "Run" button.

The code editor contains the following Kotlin code:

```
* You can edit, run, and share this code.
* play.kotlinlang.org
*/
fun main() {
    val x = 3
    when (x) {
        2, 3, 5, 7 -> println("x is a prime number between 1 and 10.")
        else -> println("x isn't a prime number between 1 and 10.")
    }
}
```

The output window below the code shows the result: "x is a prime number between 1 and 10.".

A cookie consent dialog box is visible on the right side of the screen, containing text about data collection and user consent, with "[A]ccept All" and "[M]anage Settings" buttons.

The screenshot shows the Kotlin Playground interface. At the top, there's a navigation bar with tabs for 'Course: PENGATURCARAAN MUDA' and 'Kotlin Playground: Edit, Run, Share'. Below the navigation bar is a toolbar with icons for back, forward, search, and file operations. The main area has tabs for 'Solutions', 'Docs', 'Community', 'Teach', and 'Play'. A search bar is also present. The code editor window displays a Java code snippet:

```
1.9.10 ▾ JVM ▾ Program arguments
```

```
fun main() {
    val x = 4
    when (x) {
        2, 3, 5, 7 -> println("x is a prime number between 1 and 10.")
        in 1..10 -> println("x is a number between 1 and 10, but not a prime number.")
        else -> println("x isn't a prime number between 1 and 10.")
    }
}
```

The output window shows the result: "x is a number between 1 and 10, but not a prime number."

A tooltip from the JetBrains cookie consent banner is visible in the bottom right corner:

Our website uses some cookies and records your IP address for the purposes of accessibility, security, and managing your access to the telecommunication network. You can disable data collection and cookies by changing your browser settings, but it may affect how this website functions. [Learn more](#). With your consent, JetBrains may also use cookies and your IP address to collect individual statistics and provide you with personalized offers and ads subject to the [Privacy Policy](#) and the [Terms of Use](#). JetBrains may use third-party services for this purpose. You can adjust or withdraw your consent at any time by visiting the [Opt-out page](#).

[A]ccept All [M]anage Settings

Course: PENGATURCARAAN MUDA

Kotlin Playground: Edit, Run, Share

play.kotlinlang.org/eyJ2ZXJzaW9uojIzM5LjEwliicGxhdGZvcm0iOiJqYXZhliwiYXJnci6lislm5vbVNYXJrZXljp0cnVlCj0aGVtZSI6ImlkZWEiLCjb2RljoIL...

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
/*
fun main() {
    val x: Any = 20
    when (x) {
        2, 3, 5, 7 -> println("x is a prime number between 1 and 10.")
        in 1..10 -> println("x is a number between 1 and 10, but not a prime number.")
        is Int -> println("x is an integer number, but not between 1 and 10.")
        else -> println("x isn't an integer number.")
    }
}
```

x is an integer number, but not between 1 and 10.

Our website uses some cookies and records your IP address for the purposes of accessibility, security, and managing your access to the telecommunication network. You can disable data collection and cookies by changing your browser settings, but it may affect how this website functions. [Learn more](#).

With your consent, JetBrains may also use cookies and your IP address to collect individual statistics and provide you with personalized offers and ads subject to the [Privacy Policy](#) and the [Terms of Use](#). JetBrains may use third-party services for this purpose. You can adjust or withdraw your consent at any time by visiting the [Opt-Out](#) page.

[A]ccept All [M]anage Settings

26°C Partly cloudy Q Search ENG US 1:41 AM 10/11/2023

Kotlin Playground: Edit, Run, Share

play.kotlinlang.org/eyJ2ZXJzaW9uojIzM5LjEwliicGxhdGZvcm0iOiJqYXZhliwiYXJnci6lislm5vbVNYXJrZXljp0cnVlCj0aGVtZSI6ImlkZWEiL...

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
/*
fun main() {
    val trafficLightColor = "Amber"

    when (trafficLightColor) {
        "Red" -> println("Stop")
        "Yellow", "Amber" -> println("Slow")
        "Green" -> println("Go")
        else -> println ("Invalid traffic-Light color")
}
```

Slow

Weather alert In effect Q Search ENG US 1:43 AM 10/11/2023

USE IF/ELSE AND WHEN AS EXPRESSIONS

A screenshot of the Kotlin Playground interface. The code editor contains the following Kotlin code:

```
fun main() {
    val trafficLightColor = "Black"
    val message =
        if (trafficLightColor == "Red") "Stop"
        else if (trafficLightColor == "Yellow") "Slow"
        else if (trafficLightColor == "Green") "Go"
        else "Invalid traffic-light color"
    println(message)
}
```

The output window shows the result of running the code: "Invalid traffic-light color". The playground interface includes tabs for Solutions, Docs, Community, Teach, Play, and a search bar. The status bar at the bottom shows system information like weather (26°C, Partly cloudy), date (10/11/2023), and time (1:49 AM).

A second screenshot of the Kotlin Playground interface, identical to the first one except for the code being displayed. The code is the same as above, demonstrating the use of if/else and when expressions.

Kotlin fundamentals | Android Bas... X +

developer.android.com/courses/pathways/android-basics-compose-unit-2-pathway-1#codelab-https://developer.android.com/codelabs/basic-andro...

Developers Essentials Design & Plan Develop Google Play Search English Android Studio

1 Write conditionals in Kotlin Codelab

2 Use nullability in Kotlin Codelab

3 Use classes and objects in Kotlin Codelab

4 Use function types and lambda expressions in Kotlin Codelab

5 Practice: Kotlin Fundamentals Codelab

26°C Partly cloudy Q Search Ps AI 10:55 AM 10/11/2023 ENG US

4. USE NULLABLE IN KOTLIN

USE NULLABLE VARIABLES

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    val favoriteActor = null
    println(favoriteActor)
}
```

The output window displays the result:

```
null
```

The system tray at the bottom shows a weather icon for 26°C and a date/time of 10/11/2023.

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    var favoriteActor: String = "Sandro Oh"
    favoriteActor = null
}
```

An error message is displayed in the output window:

```
① Null can not be a value of a non-null type String
```

The system tray at the bottom shows a weather icon for 26°C and a date/time of 10/11/2023.

Kotlin Playground: Edit, Run, Share

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    var favoriteActor: String? = "Sandro Oh"
    println(favoriteActor)

    favoriteActor = null
    println(favoriteActor)
}
```

Sandro Oh
null

26°C Partly cloudy Q Search ENG US 10/11/2023 10:21 AM

Kotlin Playground: Edit, Run, Share

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    var number: Int? = 10
    println(number)
}
```

10

26°C Partly cloudy Q Search ENG US 10/11/2023 10:22 AM

Kotlin Playground: Edit, Run, Share

play.kotlinlang.org/eyJZXJzaW9uIjoiMS45LjEwLjicGxhdGZvcm0iOiJqYXZhIiwiYXJnciI6IiIsIm5vb2mVNYXJrZXJzIjp0cnVlLCJ0aGVtZSI6ImlkZWEl...

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

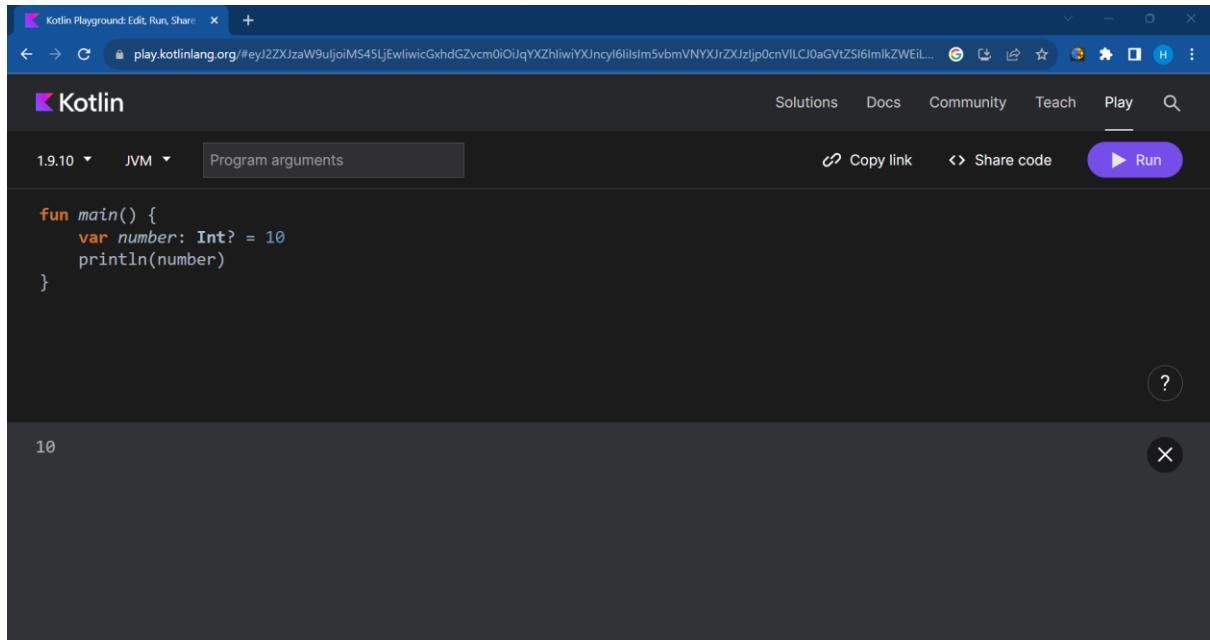
```
fun main() {
    var number: Int? = 10
    println(number)
}
```

10

26°C Partly cloudy Q Search ENG US 10/11/2023 2:02 AM

The screenshot shows the Kotlin Playground web application. The code editor contains a simple main function that prints the value of a nullable integer variable 'number' set to 10. The output window below the editor shows the result '10'. The top navigation bar includes links for Solutions, Docs, Community, Teach, and Play. Below the navigation bar are buttons for Copy link, Share code, and Run. The bottom of the screen features a Windows-style taskbar with various pinned icons and system status information.

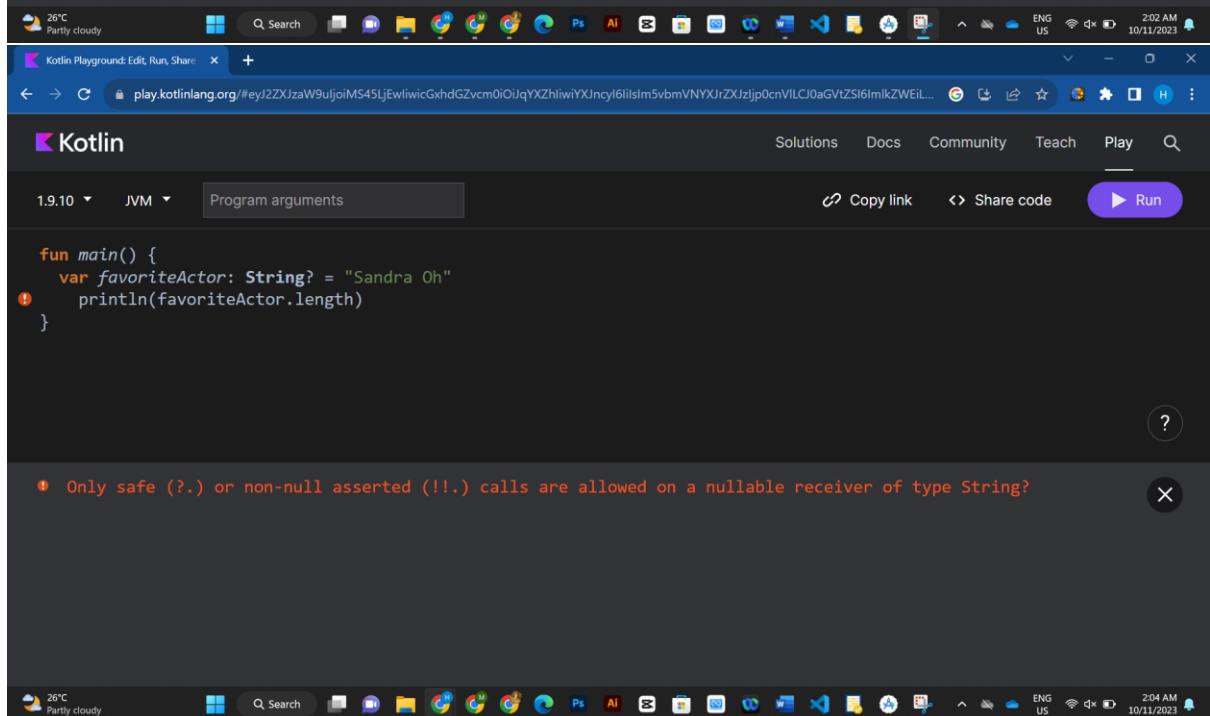
HANDLE NULLABLE VARIABLES



The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    var number: Int? = 10
    println(number)
}
```

The output window displays the result: 10.



The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    var favoriteActor: String? = "Sandra Oh"
    println(favoriteActor.length)
}
```

An error message is displayed in the output window: "Only safe (?.) or non-null asserted (!!.) calls are allowed on a nullable receiver of type String?".

A screenshot of the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    var favoriteActor: String? = "Sandra Oh"
    println(favoriteActor?.length)
}
```

The output window shows the result: 9. The status bar at the bottom indicates it's 2:05 AM on 10/11/2023.

A screenshot of the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    var favoriteActor: String? = null
    println(favoriteActor?.length)
}
```

The output window shows the result: null. The status bar at the bottom indicates it's 2:05 AM on 10/11/2023.

Kotlin Playground: Edit, Run, Share

play.kotlinlang.org/eyJ2ZXJzaW9uIjoiMS45LjEwliicGxhdGZvcmlOiiqYXZhiwiYXJncyl6lilsIm5vbmVNYXIrZXJzIjp0cnVILCJ0aGVtZSI6ImlkZWEl...

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    var favoriteActor: String? = "Sandra Oh"
    println(favoriteActor!!.length)
}
```

9

26°C Partly cloudy Q Search ENG US 10/11/2023 10:06 AM

Kotlin Playground: Edit, Run, Share

play.kotlinlang.org/eyJ2ZXJzaW9uIjoiMS45LjEwliicGxhdGZvcmlOiiqYXZhiwiYXJncyl6lilsIm5vbmVNYXIrZXJzIjp0cnVILCJ0aGVtZSI6ImlkZWEl...

Kotlin

Solutions Docs Community Teach Play

1.9.10 JVM Program arguments

Copy link Share code Run

```
fun main() {
    var favoriteActor: String? = null
    println(favoriteActor!!.length)
}
```

Exception in thread "main" java.lang.NullPointerException
at FileKt.main (File.kt:3)
at FileKt.main (File.kt:-1)
at jdk.internal.reflect.NativeMethodAccessorImpl.invoke0 (:2)

26°C Partly cloudy Q Search ENG US 10/11/2023 10:07 AM

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    var favoriteActor: String? = "Sandra Oh"

    if(favoriteActor != null){
        println("The number of characters in your favourite actor's name is ${favoriteActor.length}.")
    }
}
```

The output window displays the result of the execution:

```
The number of characters in your favourite actor's name is 9.
```

The status bar at the bottom shows the weather as 26°C Partly cloudy, the date and time as 10/11/2023 2:09 AM, and system status like ENG US.

The screenshot shows the Kotlin Playground interface. The code in the editor is:

```
fun main() {
    var favoriteActor: String? = null

    if(favoriteActor != null){
        println("The number of characters in your favourite actor's name is ${favoriteActor.length}.")
    } else {
        println("You didn't input a name.")
    }
}
```

The output window displays the result of the execution:

```
You didn't input a name.
```

The status bar at the bottom shows the weather as 26°C Partly cloudy, the date and time as 10/11/2023 2:10 AM, and system status like ENG US.

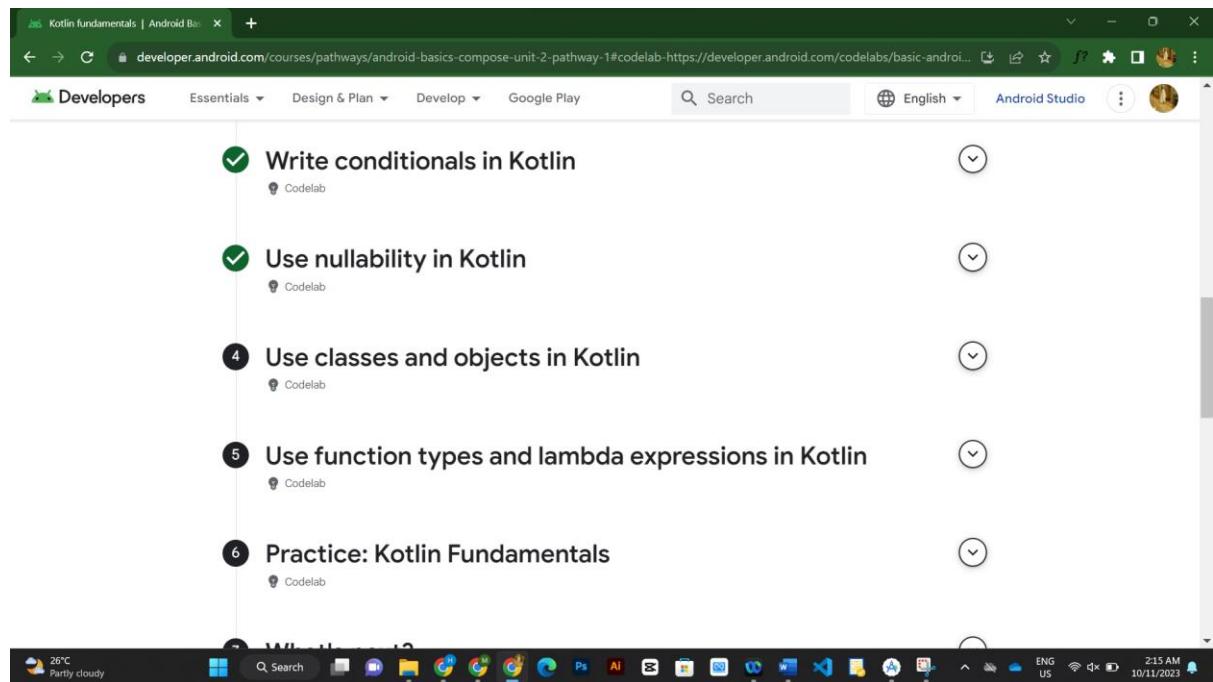
The image shows a Windows desktop environment with two identical instances of the Kotlin Playground web application open in separate browser windows. Both windows are titled "Kotlin" and show the same code snippet. The code defines a main function that prints the length of a favorite actor's name. In the first window, the variable declaration uses the standard syntax `val`. In the second window, it uses the nullable variable declaration syntax `val` followed by a question mark and a colon. Both windows display the output "The number of characters in your fav actor's name is 9." at the bottom of the code editor.

```
fun main() {
    var favoriteActor: String? = "Sandra Oh"

    val LengthOfName = if (favoriteActor != null) {
        favoriteActor.length
    } else {
        0
    }

    println("The number of characters in your fav actor's name is $lengthOfName.")

    The number of characters in your fav actor's name is 9.
```



4.

The screenshot shows a web browser window with the URL developer.android.com/courses/pathways/android-basics-compose-unit-2-pathway-1#codelab. The page displays a list of codelabs under the 'Kotlin fundamentals' category:

- 1 Use classes and objects in Kotlin (Completed)
- 5 Use function types and lambda expressions in Kotlin
- 6 Practice: Kotlin Fundamentals
- 7 What's next? (Video Optional)

Below the list, there is a section titled "Quiz". The system tray at the bottom shows the date as 15/11/2023 and the time as 8:48 PM.

The screenshot shows a web browser window with the URL [play.kotlinlang.org](https://play.kotlinlang.org/#eyjZZXJzaW9uJoiMS45LjIwIicGxhdGZvcm0iOjUqYXZhwiYXJncyl6lislm5vbmVNXYJrZXJzlp0cnVLCj0aGVtZSI6ImkZWEiLCjb2Rljoia...). The page displays the following Kotlin code:

```
import kotlin.properties.ReadWriteProperty
import kotlin.reflect.KProperty

open class SmartDevice(val name: String, val category: String) {

    var deviceStatus = "online"
        protected set

    open val deviceType = "unknown"

    Android TV is turned on. Speaker volume is set to 2 and channel number is s
    Google Light turned on. The brightness level is 2.
}
```

A cookie consent dialog is visible on the right side of the screen, containing text about data collection and processing, and two buttons: "[A]ccept All" and "[M]anage Settings". The system tray at the bottom shows the date as 15/11/2023 and the time as 8:46 PM.

5.

The screenshot shows a web browser window with the URL developer.android.com/courses/pathways/android-basics-compose-unit-2-pathway-1#codelab. The page displays a codelab titled "Kotlin Fundamentals" with four steps: "Use classes and objects in Kotlin", "Use function types and lambda expressions in Kotlin", "Practice: Kotlin Fundamentals", and "What's next?". Below the codelab is a "Quiz" section with the instruction "Test what you've learned and earn your Kotlin fundamentals badge." A terminal window is overlaid on the browser, showing the output of a Kotlin program. The terminal window title is "Kotlin" and it shows the following code and output:

```
fun main() {
    val numberOfQuarters = 5

    repeat(numberOfQuarters) {
        println("Have a treat!")
    }

    if (numberOfQuarters == 0) {
        println("No treats!")
    }
}

Have a treat!
```

6.

The screenshot shows a web browser window with the URL developer.android.com/courses/pathways/android-basics-compose-unit-2-pathway-1#codelab. The page title is "Practice: Kotlin Fundamentals". Below it is a section titled "What's next?" with a "Video" link. A large box labeled "Quiz" contains the text "Test what you've learned and earn your Kotlin fundamentals badge." and a "Take the quiz" button. At the bottom of the page, there is a "Was this helpful?" feedback section with thumbs up and down icons.

The screenshot shows the Kotlin Playground interface at play.kotlinlang.org. The code entered is:

```
open class Phone(var isScreenLightOn: Boolean = false){
    open fun switchOn() {
        isScreenLightOn = true
    }

    fun switchOff() {
        isScreenLightOn = false
    }

    fun checkPhoneScreenLight() {
        if (isScreenLightOn) {
            println("The phone screen's light is on.")
        } else {
            println("The phone screen's light is off.")
        }
    }
}
```

The output window displays the results of running the code:

```
The phone screen's light is off.
The phone screen's light is on.
```

QUIZ TASK 1

Kotlin fundamentals | Android Dev ChatGPT

developer.android.com/courses/quizzes/android-basics-compose-unit-2-pathway-1/android-basics-compose-unit-2-pathway-1?continue=https%3A%...

Developers Essentials Design & Plan Develop Google Play Search English Android Studio Trailing lambda

Results

You scored **8 out of 10**. Congratulations! You have passed this quiz.

You earned the **Kotlin Fundamentals badge!**

The badge has been added to your profile.

Share  Return to pathway View profile

Was this helpful? 

USD/MYR -0.98% Search  ENG US 9:23 PM 15/11/2023

TASK 2:

Create an interactive Dice Roller app

3 Use the debugger in Android Studio

4 Practice: Click behavior

5 What's next?

Quiz

Test your knowledge and earn your Dice Roller badge.

Dice Roller - MainActivity.kt [Dice_Roller.app.main]

```
1 package com.example.diceroller
2
3 import ...
4
5 class MainActivity : ComponentActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContent {
9             DiceRollerTheme {
10                 Surface(
11                     modifier = Modifier.fillMaxSize(),
12                     color = MaterialTheme.colorScheme.background
13                 ) {
14                     DiceRollerApp()
15                 }
16             }
17         }
18     }
19
20     @Preview
21     @Composable
22     fun DiceRollerApp() {
23         DiceWithButtonAndImage(modifier = Modifier
24             .fillMaxSize()
25             .wrapContentSize(Alignment.Center)
26         )
27     }
28 }
```

DiceRollerApp

DiceWithButtonAndImage

Compose Preview refresh: refreshing Compose previews total elapsed time was 0 s 054 ms (moments ago)

The screenshot shows a web browser window with several tabs open. The active tab displays a codelab titled "Create an interactive Dice Roller app" from Google Developers. The codelab includes sections for "Use the debugger in Android Studio", "Practice: Click behavior", and "What's next?". Below the codelab, there is a "Quiz" section with the heading "Quiz" and the sub-instruction "Test your knowledge and earn your Dice Roller badge.".

The screenshot shows the Android Studio interface with the "Dice Roller" project open. The "Run" menu is currently open, displaying various options for running and debugging the application. The option "Attach to Process..." is highlighted with a red circle, indicating it is the focus of the current step. The "Gradle" tab is selected in the top navigation bar.

The screenshot shows a web browser window with several tabs open. The main content area displays a list of codelabs from the Android Developers website:

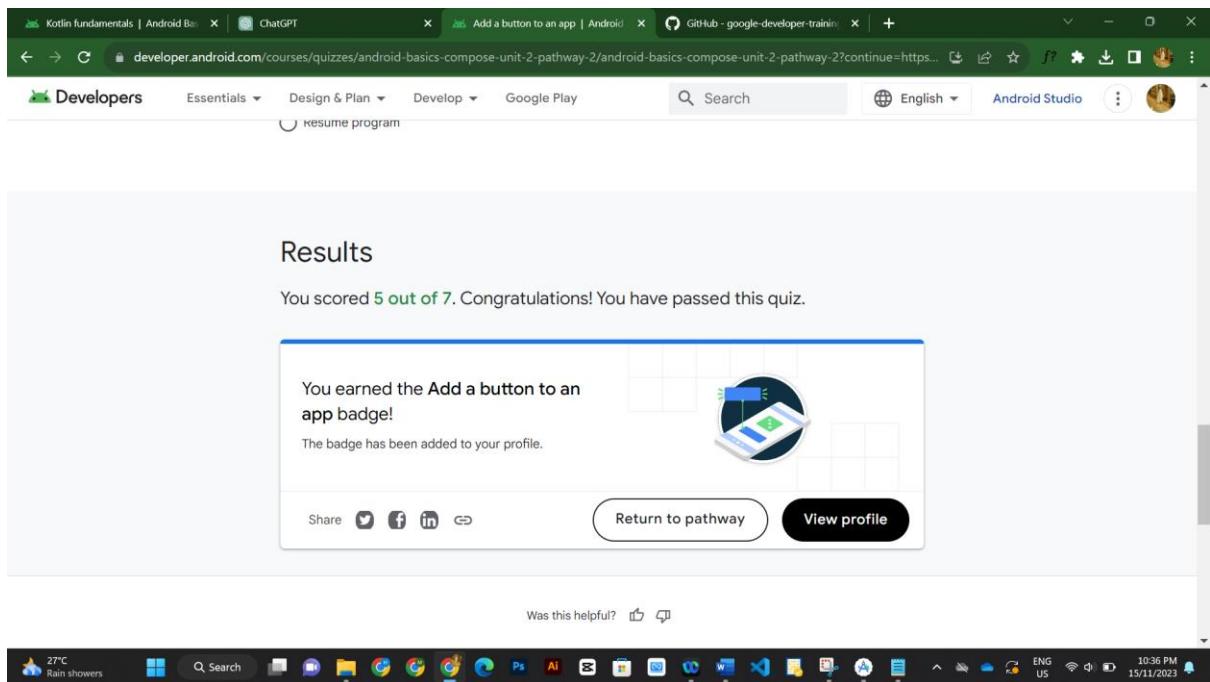
- ✓ Use the debugger in Android Studio
- ✓ Practice: Click behavior
- 5 What's next?

Below this list is a section titled "Quiz" with the sub-instruction "Test your knowledge and earn your Dice Roller badge." A "Take the quiz" button is visible.

The screenshot shows the Android Studio interface. On the left is the Project Structure sidebar, which lists files like build.gradle.kts, MainActivity.kt, and themes.xml. The main editor pane contains Java code for a theme:

```
79     outlineVariant = md_theme_dark_outlineVariant,
80
81     )
82     @Composable
83     fun AppTheme(
84         darkTheme: Boolean = isSystemInDarkTheme(),
85         dynamicColor: Boolean = false,
86         content: @Composable () -> Unit
87     ) {
88         val colorScheme = when {
89             dynamicColor && Build.VERSION.SDK_INT >= Build.VERSION_CODES.S
90             val context = LocalContext.current
91             if (darkTheme) dynamicDarkColorScheme(context) else dynamicLightColorScheme(context)
92         }
93
94         darkTheme -> DarkColorScheme
95         else -> LightColorScheme
96     }
97
98     val view = LocalView.current
99     if (!view.isInEditMode) {
100         SideEffect {
101             val window = (view.context as Activity).window
102             window.statusBarColor = colorScheme.primary.toArgb()
103             WindowCompat.getInsetsController(window, view).isAppearanceLightStatusBars = true
104         }
105     }
106
107     MaterialTheme(
108         colorScheme = colorScheme,
109         typography = Typography,
110         content = content
111     )
112 }
```

To the right of the editor is a preview window showing a smartphone screen with a yellow lemon icon and the text "Keep tapping the lemon to square it". The bottom status bar of the OS shows the date and time as 15/11/2023 and 10:31 PM.



GITHUB LINK:

<https://github.com/HazimShakri7/NativeLab>