Beginner Level for Android Development using Kotlin

Instructor: Ferry Yuwono

Module 1:
Introduction to Android and Kotlin

1.1 Introduction + Course Outline

Introduction to Android and Kotlin

Kotlin is a modern statically typed programming language used by **over 60% of professional Android developers** that helps boost productivity, developer satisfaction, and code safety.

Since the release of **Android Studio 3.0** in **October 2017**, **Kotlin** has been included as an alternative to the standard Java compiler

On **7 May 2019**, Google announced that the **Kotlin** programming language is now its preferred language for **Android** app developers

1.1 Introduction + Course Outline

Course Outline

Module 1: Introduction to Android and Kotlin

1.1 Introduction + Course Outline
1.2 Install Android Studio
1.3 Create First Android Application
1.4 Setup Android Emulator

$\rightarrow$

1.5 Kotlin Language
1.6 Declare Variables and Data Types
1.7 Array and Collections
1.8 Conditions
1.9 Loops
1.10 Functions
1.11 Class and Data Class

1.1 Introduction + Course Outline

Course Outline

Module 2: Develop Your First Android Application

2.1 Android Project Structure
2.2 Layouts and TextView
2.3 EditText and Button
2.4 ImageView and ImageButton
2.5 View Binding

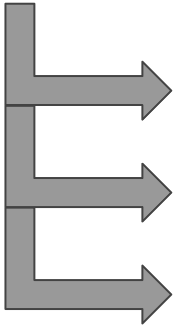
→

2.6 Persistent data
2.7 Activity and Manifest
2.8 RecyclerView and Adapter
2.9 Final Assignment

1.1 Introduction + Course Outline

Repository

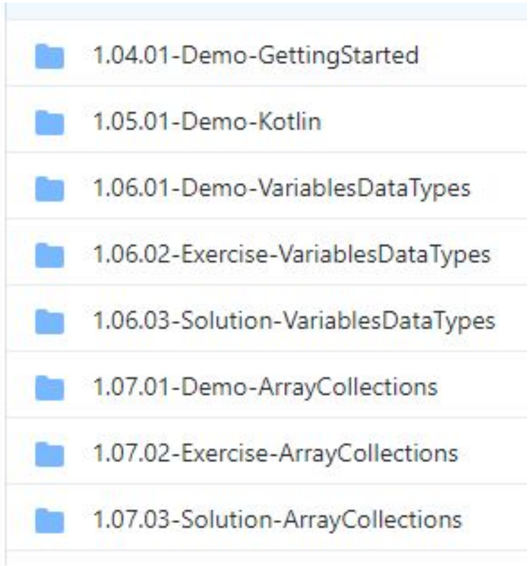
https://github.com/ferryyuwono/android-beginner



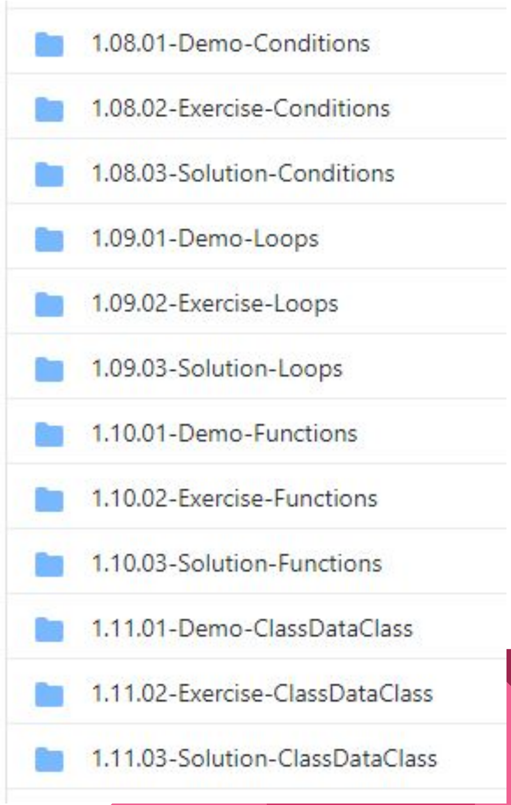
Demo Code

Exercise Code

Solution Code



1.04.01-Demo-GettingStarted
1.05.01-Demo-Kotlin
1.06.01-Demo-VariablesDataTypes
1.06.02-Exercise-VariablesDataTypes
1.06.03-Solution-VariablesDataTypes
1.07.01-Demo-ArrayCollections
1.07.02-Exercise-ArrayCollections
1.07.03-Solution-ArrayCollections

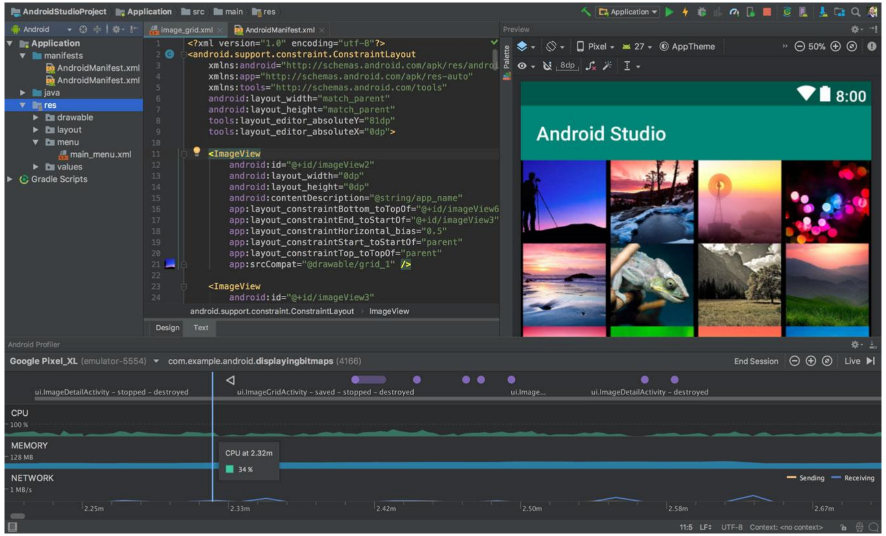


1.08.01-Demo-Conditions
1.08.02-Exercise-Conditions
1.08.03-Solution-Conditions
1.09.01-Demo-Loops
1.09.02-Exercise-Loops
1.09.03-Solution-Loops
1.10.01-Demo-Functions
1.10.02-Exercise-Functions
1.10.03-Solution-Functions
1.11.01-Demo-ClassDataClass
1.11.02-Exercise-ClassDataClass
1.11.03-Solution-ClassDataClass

1.2 Install Android Studio

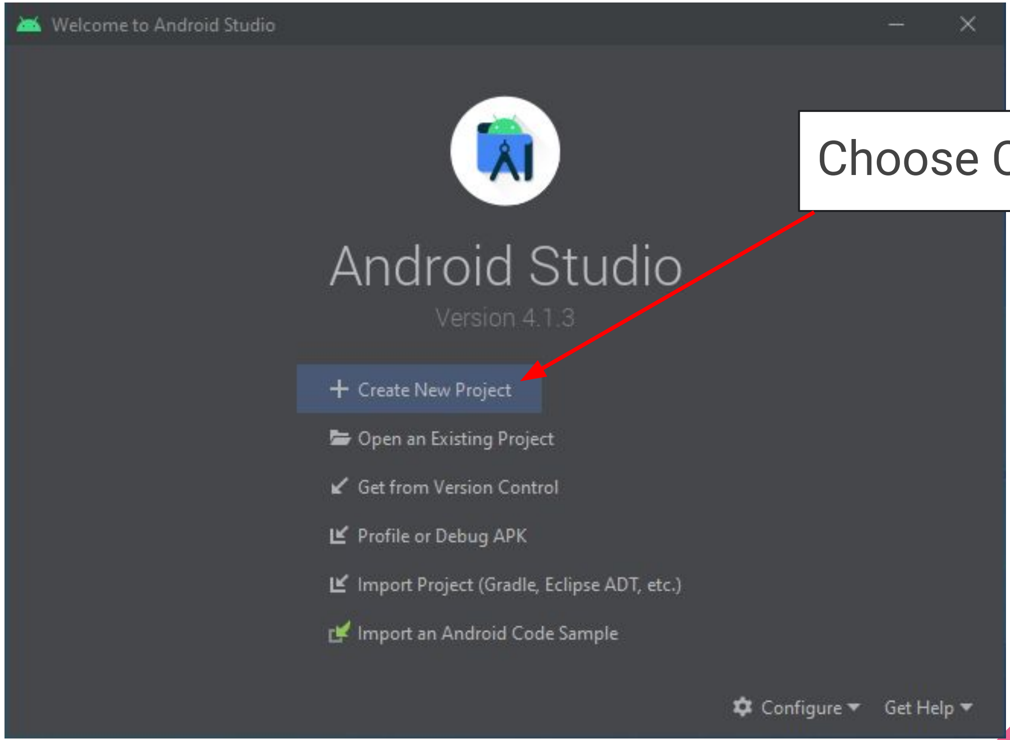
Download Android Studio

https://developer.android.com/studio



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteY="81dp"
    tools:layout_editor_absoluteX="0dp">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:contentDescription="@string/app_name"
        app:layout_constraintBottom_toTopOf="@+id/imageView6
        app:layout_constraintEnd_toStartOf="@+id/imageView3"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/grid_1" />

    <ImageView
        android:id="@+id/imageView3"
```

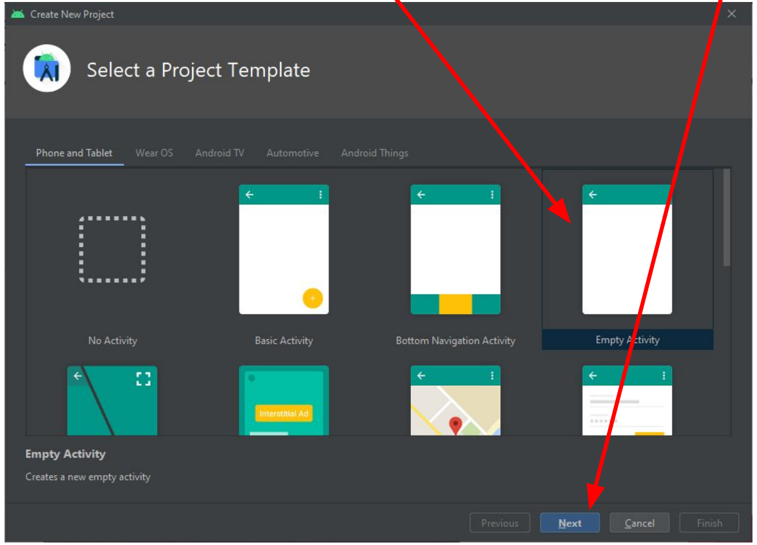
1.3 Create First Android Application



Choose Create New Project

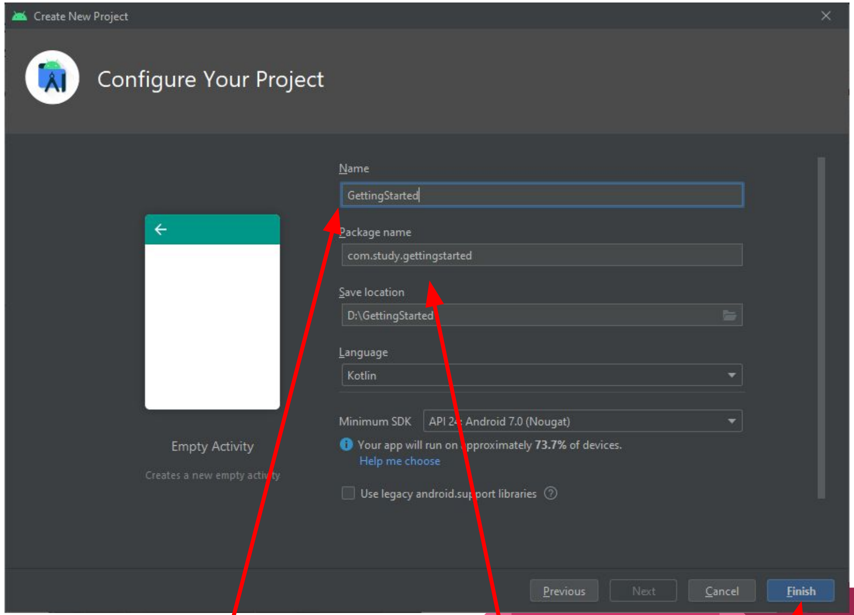
1.3 Create First Android Application

Choose Empty Activity, then Next



Empty Activity

Creates a new empty activity



Name: GettingStarted

Package name: com.study.gettingstarted

Save location: D:\GettingStarted

Language: Kotlin

Minimum SDK: API 24: Android 7.0 (Nougat)

Your app will run on approximately 73.7% of devices.

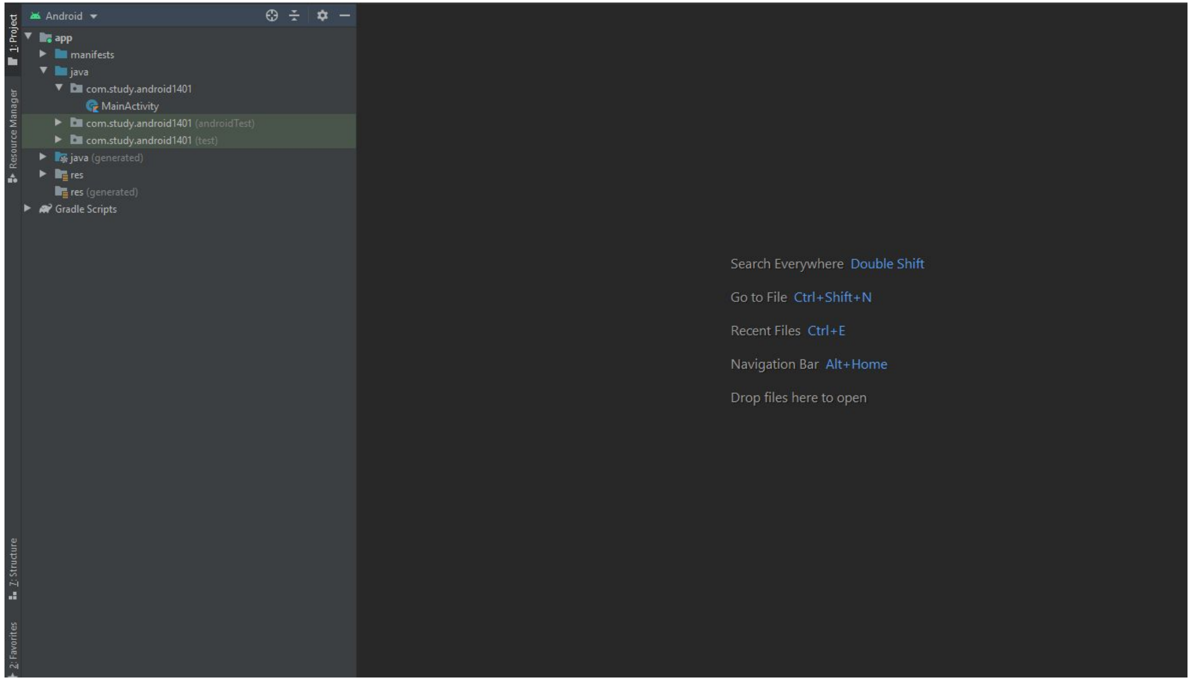
Help me choose

Use legacy android.support libraries

Fill Name, Package Name, then Finish

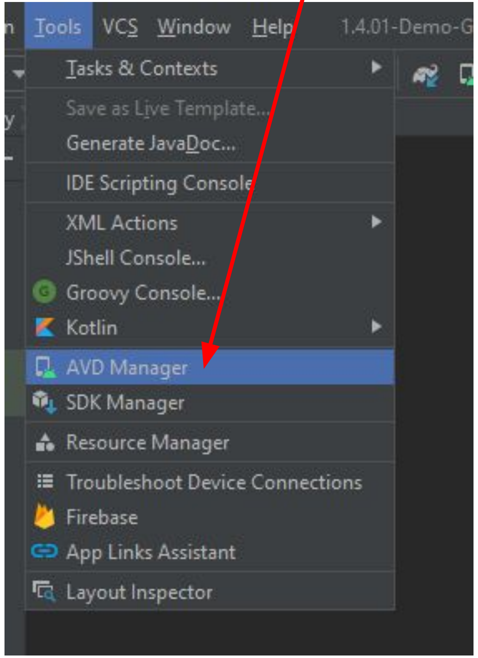
1.4 Setup Android Emulator

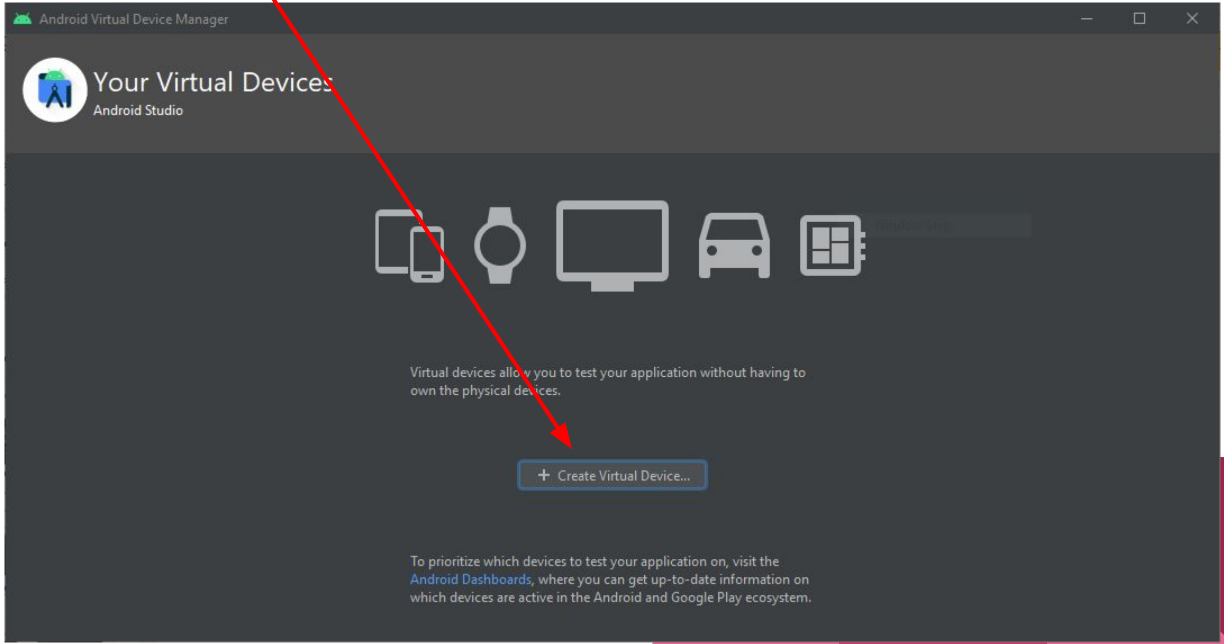
https://github.com/ferryyuwono/android-beginner/tree/main/1.04.01-Demo-GettingStarted



1.4 Setup Android Emulator

Tools > AVD Manager > Create Virtual Devices...

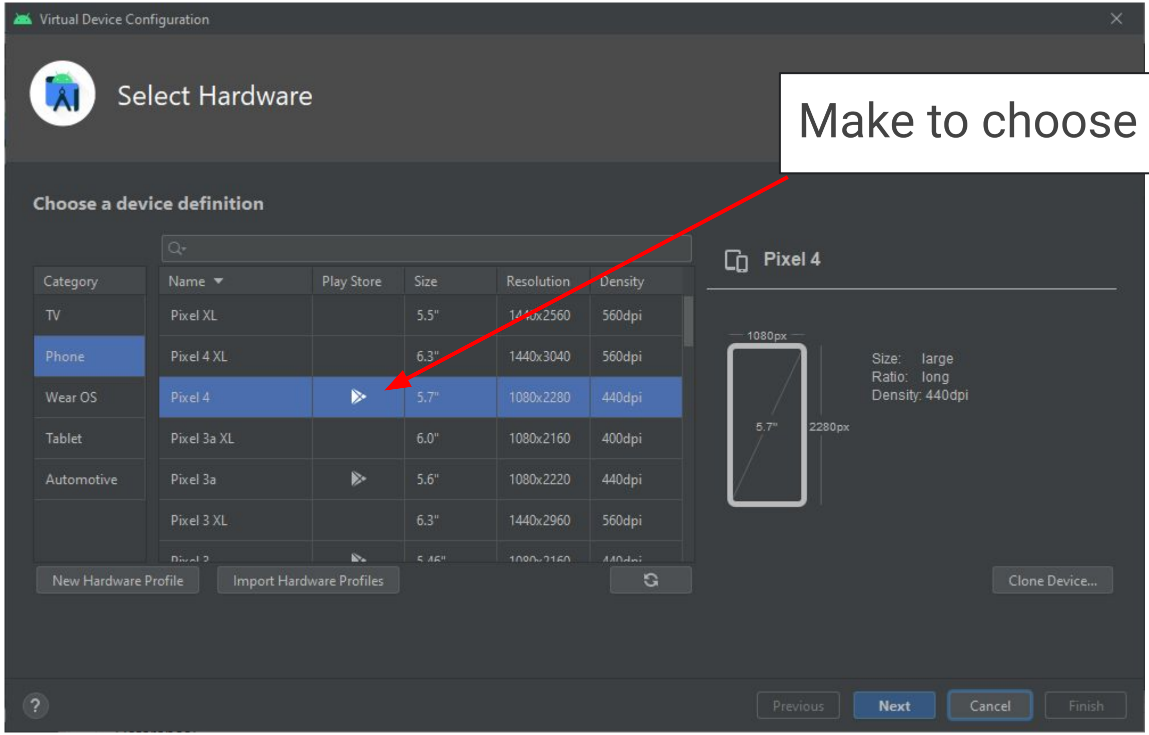




Virtual devices allow you to test your application without having to own the physical devices.

To prioritize which devices to test your application on, visit the Android Dashboards, where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

1.4 Setup Android Emulator



Choose a device definition

Category
TV
Phone
Wear OS
Tablet
Automotive

Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2560	560dpi
Pixel 4 XL		6.3"	1440x3040	560dpi
Pixel 4	▶	5.7"	1080x2280	440dpi
Pixel 3a XL		6.0"	1080x2160	400dpi
Pixel 3a	▶	5.6"	1080x2220	440dpi
Pixel 3 XL		6.3"	1440x2960	560dpi
Pixel 3	▶	5.46"	1080x2160	440dpi

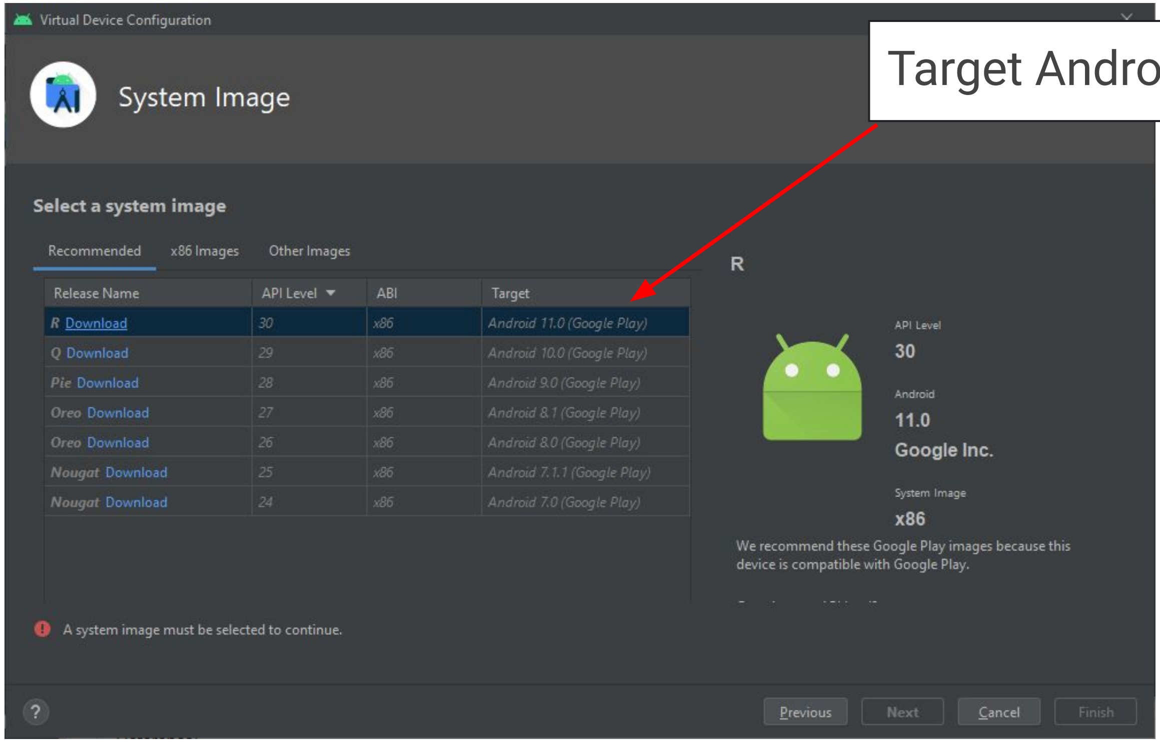
Pixel 4

1080px, 2280px, 5.7"

Size: large
Ratio: long
Density: 440dpi

Make to choose device with Play Store

1.4 Setup Android Emulator



Target Android (Google Play)

Virtual Device Configuration

System Image

Select a system image

Recommended | x86 Images | Other Images

Release Name	API Level	ABI	Target
R Download	30	x86	Android 11.0 (Google Play)
Q Download	29	x86	Android 10.0 (Google Play)
Pie Download	28	x86	Android 9.0 (Google Play)
Oreo Download	27	x86	Android 8.1 (Google Play)
Oreo Download	26	x86	Android 8.0 (Google Play)
Nougat Download	25	x86	Android 7.1.1 (Google Play)
Nougat Download	24	x86	Android 7.0 (Google Play)

R

API Level
30

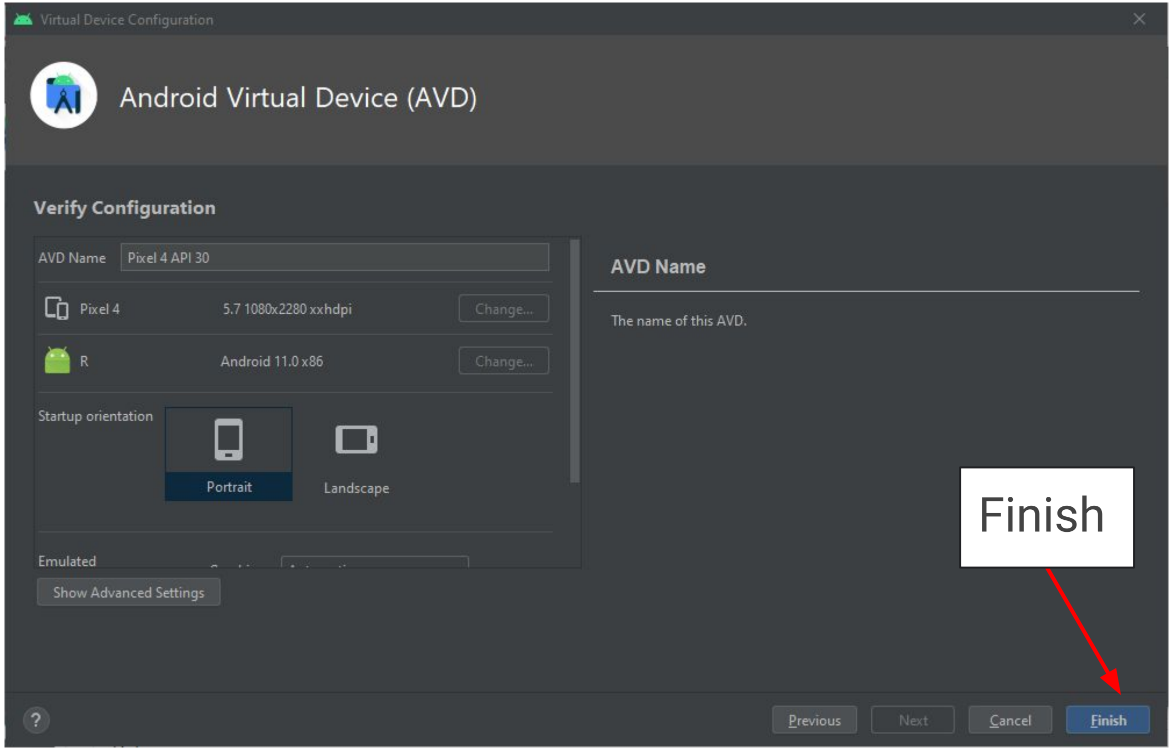
Android
11.0
Google Inc.

System Image
x86

We recommend these Google Play images because this device is compatible with Google Play.

A system image must be selected to continue.

1.4 Setup Android Emulator



Verify Configuration

AVD Name: Pixel 4 API 30

Pixel 4 — 5.7 1080x2280 xxhdpi

R — Android 11.0 x86

Startup orientation: [✓ Portrait] Landscape

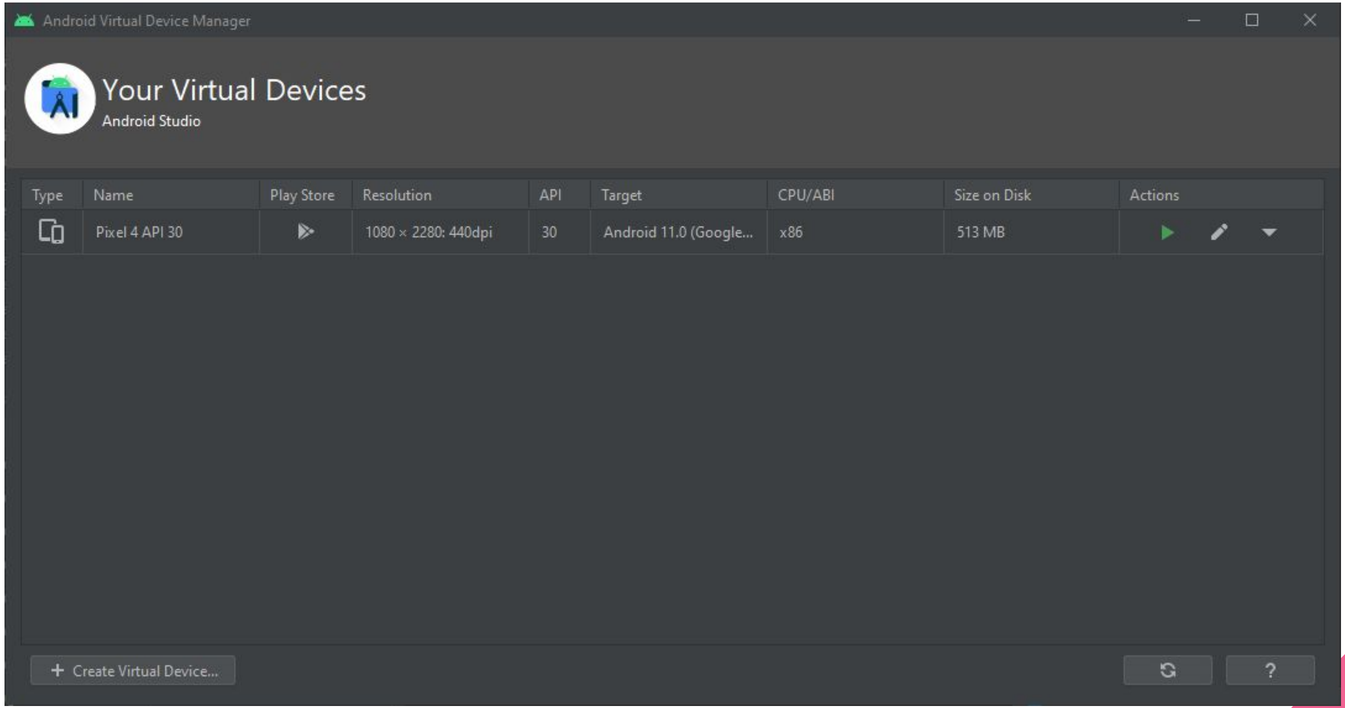
Emulated

AVD Name

The name of this AVD.

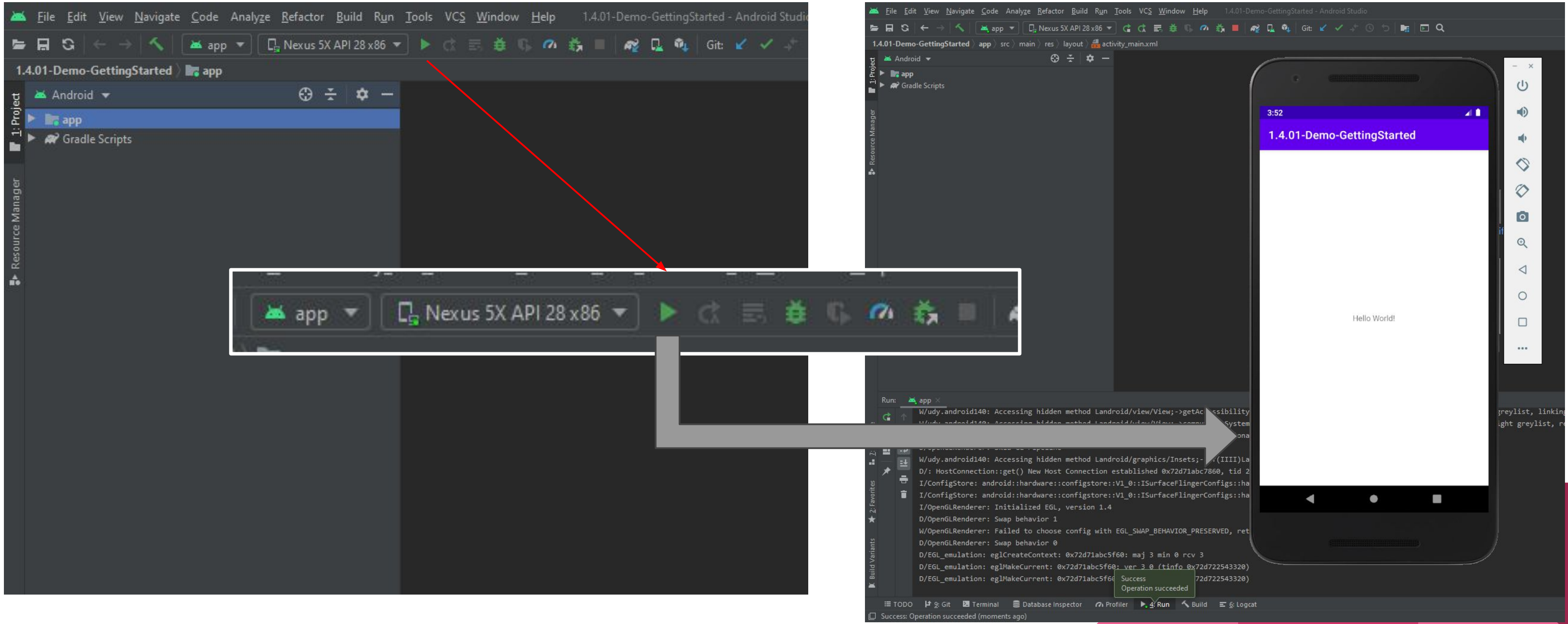
Finish

1.4 Setup Android Emulator



Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Pixel 4 API 30		1080 × 2280: 440dpi	30	Android 11.0 (Google...	x86	513 MB	

1.4 Setup Android Emulator



```
W/udy.android140: Accessing hidden method Landroid/view/View;->getAc
W/udy.android140: Accessing hidden method Landroid/graphics/Insets;-
D/: HostConnection::get() New Host Connection established 0x72d71abc7860, tid 2
I/ConfigStore: android::hardware::configstore::V1_0::ISurfaceFlingerConfigs::ha
I/ConfigStore: android::hardware::configstore::V1_0::ISurfaceFlingerConfigs::ha
I/OpenGLRenderer: Initialized EGL, version 1.4
D/OpenGLRenderer: Swap behavior 1
W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, ret
D/OpenGLRenderer: Swap behavior 0
D/EGL_emulation: eglCreateContext: 0x72d71abc5f60: maj 3 min 0 rcv 3
D/EGL_emulation: eglMakeCurrent: 0x72d71abc5f60: ver 3 0 (tinfo 0x72d722543320)
D/EGL_emulation: eglMakeCurrent: 0x72d71abc5f6                 72d722543320)
```

Success: Operation succeeded (moments ago)

1.5 Kotlin Language

https://github.com/ferryyuwono/android-beginner/tree/main/1.05.01-Demo-Kotlin

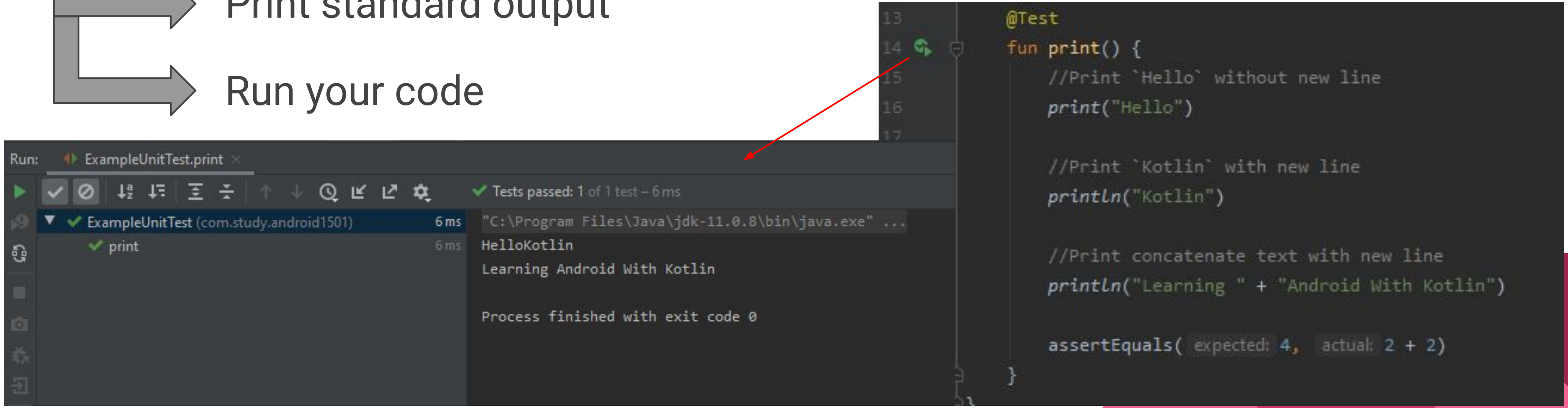
File: ExampleUnitTest.kt

- Add comment to code
- Print standard output
- Run your code

Format:

print(text to print)

println(text to print)



```kotlin
@Test
fun print() {
    //Print `Hello` without new line
    print("Hello")

    //Print `Kotlin` with new line
    println("Kotlin")

    //Print concatenate text with new line
    println("Learning " + "Android With Kotlin")

    assertEquals( expected: 4, actual: 2 + 2)
}
```

Tests passed: 1 of 1 test – 6 ms

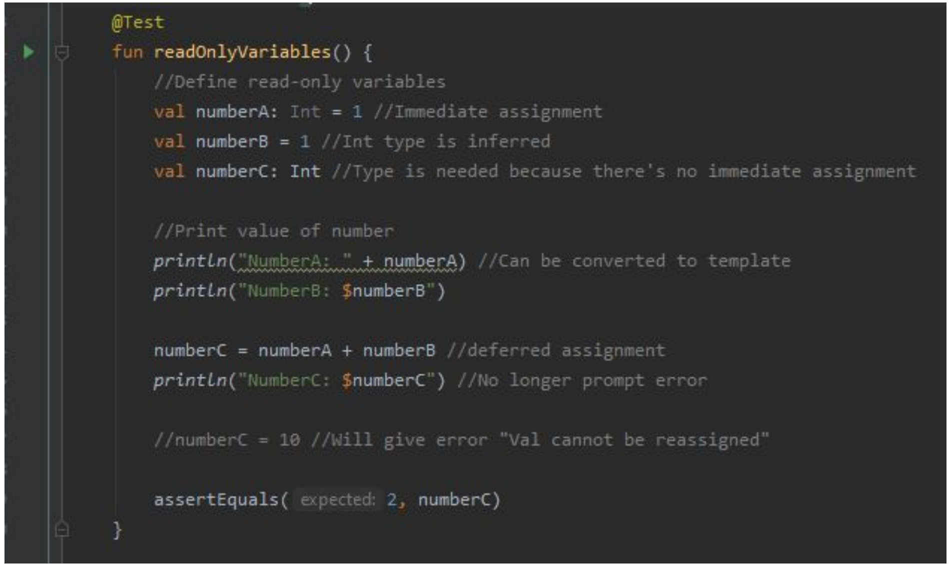
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
HelloKotlin
Learning Android With Kotlin

Process finished with exit code 0
```

1.6.1 [Demo] Declare Variables and Data Types

https://github.com/ferryyuwono/android-beginner/tree/main/1.06.01-Demo-VariablesDataTypes

File: ExampleUnitTest.kt

Read only variables

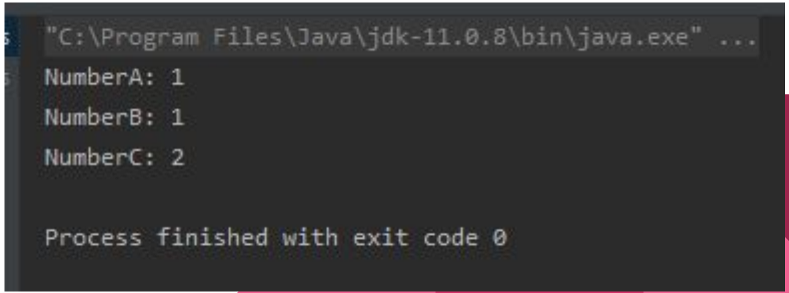


```kotlin
@Test
fun readOnlyVariables() {
    //Define read-only variables
    val numberA: Int = 1 //Immediate assignment
    val numberB = 1 //Int type is inferred
    val numberC: Int //Type is needed because there's no immediate assignment

    //Print value of number
    println("NumberA: " + numberA) //Can be converted to template
    println("NumberB: $numberB")

    numberC = numberA + numberB //deferred assignment
    println("NumberC: $numberC") //No longer prompt error

    //numberC = 10 //Will give error "Val cannot be reassigned"

    assertEquals( expected: 2, numberC)
}
```

Format:

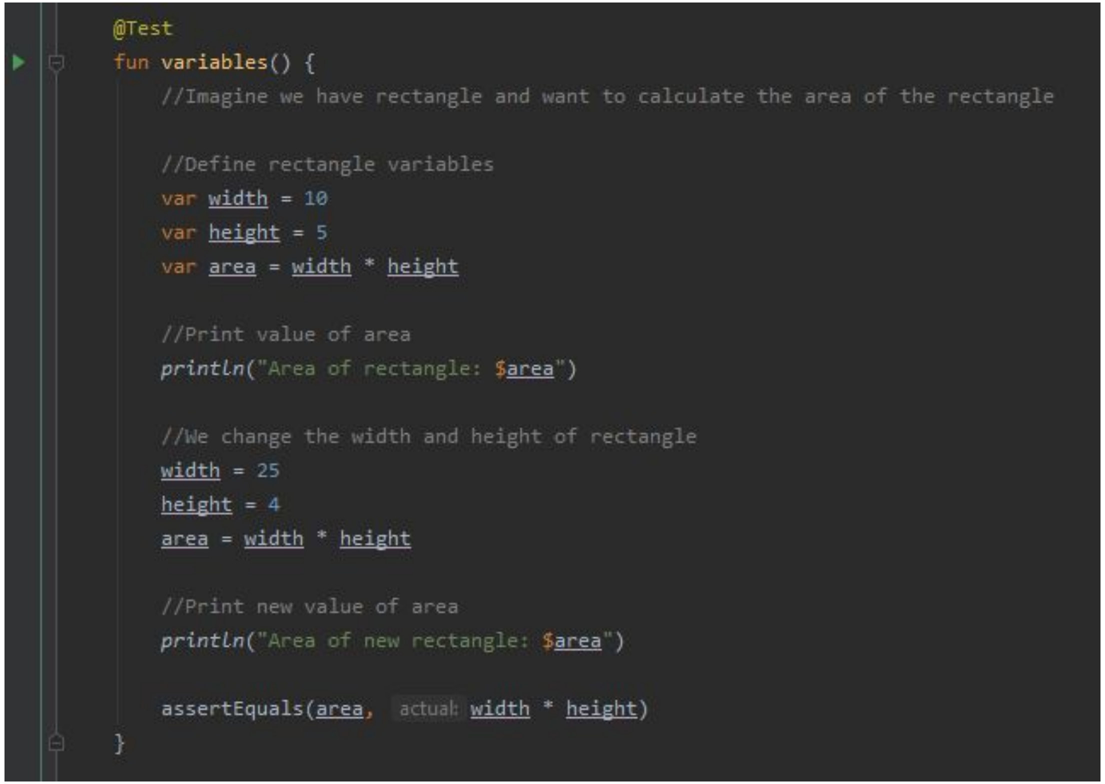
val variable name : data type

variable name: lower camelCase (start with lowercase letter and separate the words with a single capitalized letter)



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
NumberA: 1
NumberB: 1
NumberC: 2

Process finished with exit code 0
```

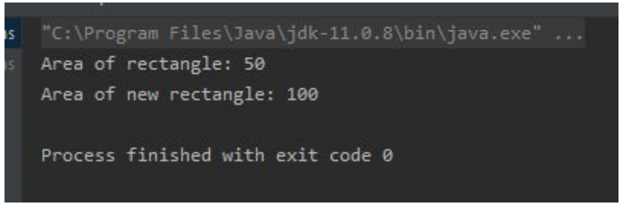
Variables



```
@Test
fun variables() {
    //Imagine we have rectangle and want to calculate the area of the rectangle

    //Define rectangle variables
    var width = 10
    var height = 5
    var area = width * height

    //Print value of area
    println("Area of rectangle: $area")

    //We change the width and height of rectangle
    width = 25
    height = 4
    area = width * height

    //Print new value of area
    println("Area of new rectangle: $area")

    assertEquals(area,  actual: width * height)
}
```

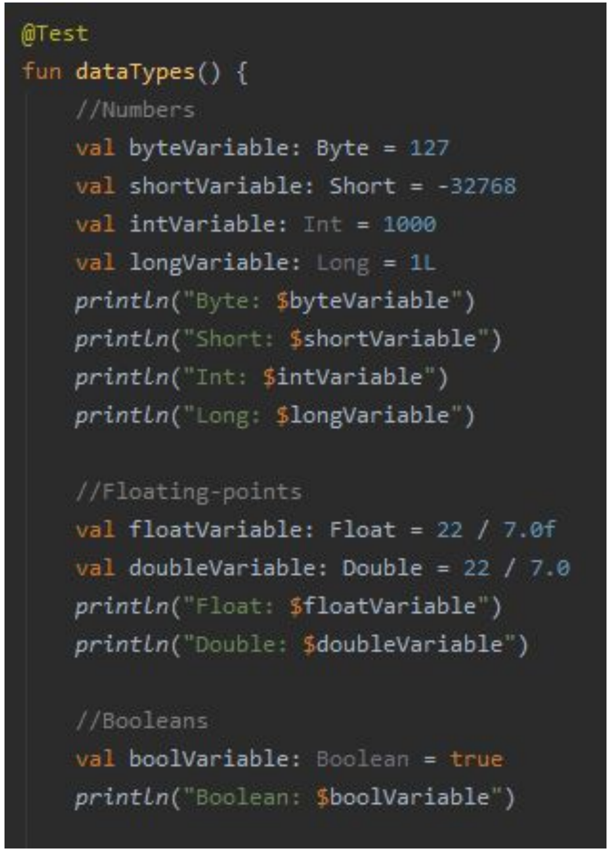
Format:

var variable name : data type

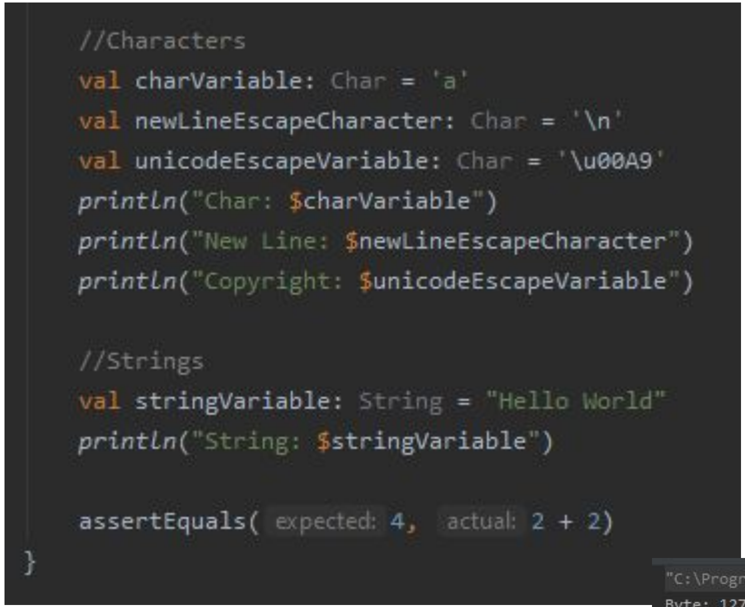


```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Area of rectangle: 50
Area of new rectangle: 100

Process finished with exit code 0
```

Data Types



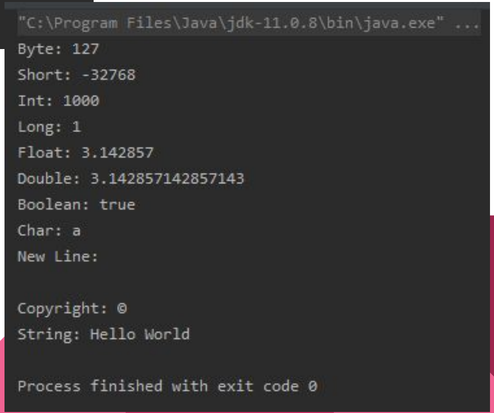
```
@Test
fun dataTypes() {
    //Numbers
    val byteVariable: Byte = 127
    val shortVariable: Short = -32768
    val intVariable: Int = 1000
    val longVariable: Long = 1L
    println("Byte: $byteVariable")
    println("Short: $shortVariable")
    println("Int: $intVariable")
    println("Long: $longVariable")

    //Floating-points
    val floatVariable: Float = 22 / 7.0f
    val doubleVariable: Double = 22 / 7.0
    println("Float: $floatVariable")
    println("Double: $doubleVariable")

    //Booleans
    val boolVariable: Boolean = true
    println("Boolean: $boolVariable")
```



```
    //Characters
    val charVariable: Char = 'a'
    val newLineEscapeCharacter: Char = '\n'
    val unicodeEscapeVariable: Char = '\u00A9'
    println("Char: $charVariable")
    println("New Line: $newLineEscapeCharacter")
    println("Copyright: $unicodeEscapeVariable")

    //Strings
    val stringVariable: String = "Hello World"
    println("String: $stringVariable")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Data Types:

Byte, Short, Int, Long
Float, Double
Boolean
Char
String



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Byte: 127
Short: -32768
Int: 1000
Long: 1
Float: 3.142857
Double: 3.142857142857143
Boolean: true
Char: a
New Line: 

Copyright: ©
String: Hello World

Process finished with exit code 0
```

1.6.2 [Exercise] Declare Variables and Data Types

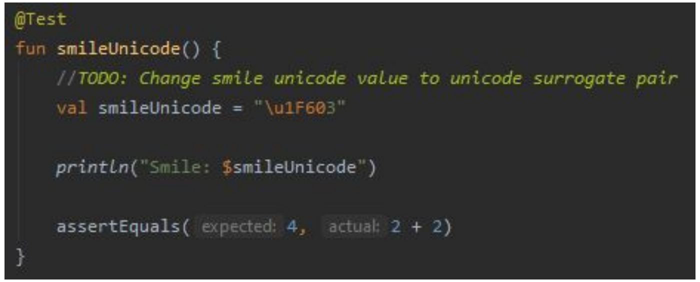
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.06.02-Exercise-VariablesDataTypes

File

ExampleUnitTest.kt

- Write simpler code
- Explicit Conversion
- Nullable data type
- Unsigned data type
- Surrogate pair unicode



```
@Test
fun smileUnicode() {
    //TODO: Change smile unicode value to unicode surrogate pair
    val smileUnicode = "\u1F603"

    println("Smile: $smileUnicode")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

1.6.3 [Solution] Declare Variables and Data Types

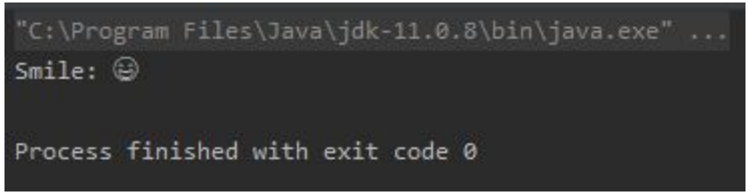
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.06.03-Solution-VariablesDataTypes

File

ExampleUnitTest.kt

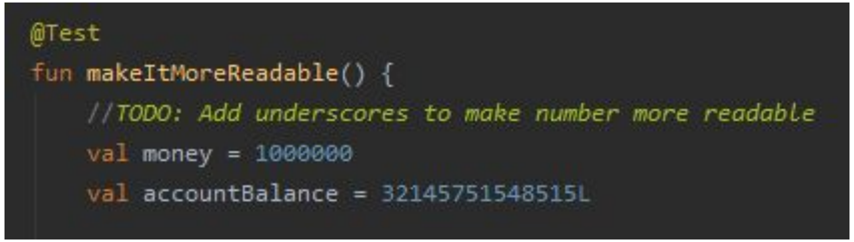
- Write simpler code
- Explicit Conversion
- Nullable data type
- Unsigned data type
- Surrogate pair unicode



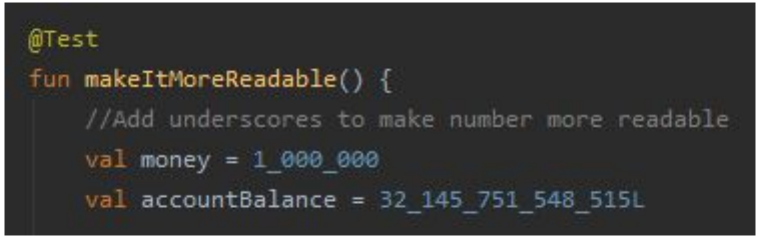
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Smile: 😆

Process finished with exit code 0
```

1.6.3 [Solution] Declare Variables and Data Types

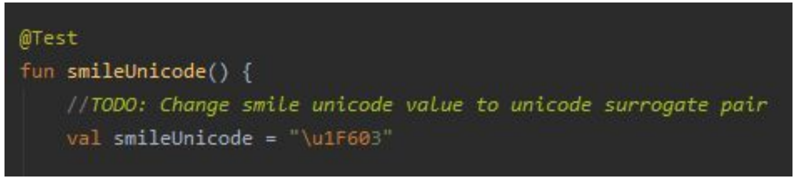
Example on how to do the exercise and compare with solution



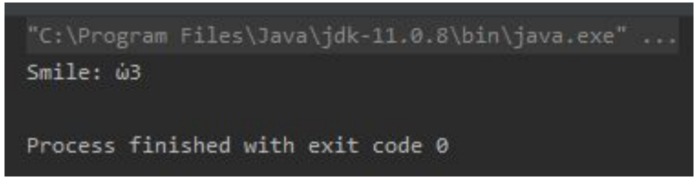
```
@Test
fun makeItMoreReadable() {
    //TODO: Add underscores to make number more readable
    val money = 1000000
    val accountBalance = 32145751548515L
```



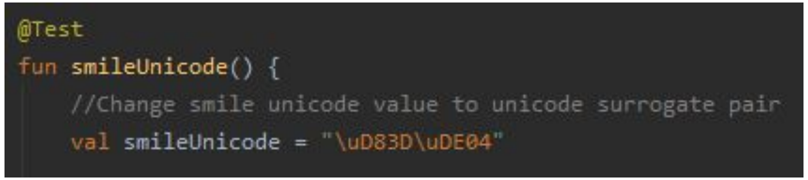
```
@Test
fun makeItMoreReadable() {
    //Add underscores to make number more readable
    val money = 1_000_000
    val accountBalance = 32_145_751_548_515L
```



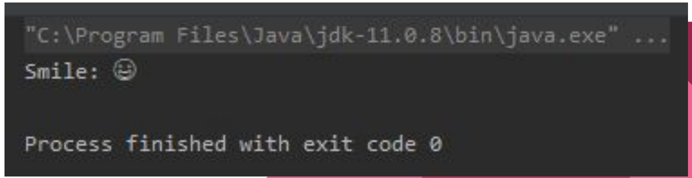
```
@Test
fun smileUnicode() {
    //TODO: Change smile unicode value to unicode surrogate pair
    val smileUnicode = "\u1F603"
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Smile: ὠ3

Process finished with exit code 0
```



```
@Test
fun smileUnicode() {
    //Change smile unicode value to unicode surrogate pair
    val smileUnicode = "\uD83D\uDE04"
```



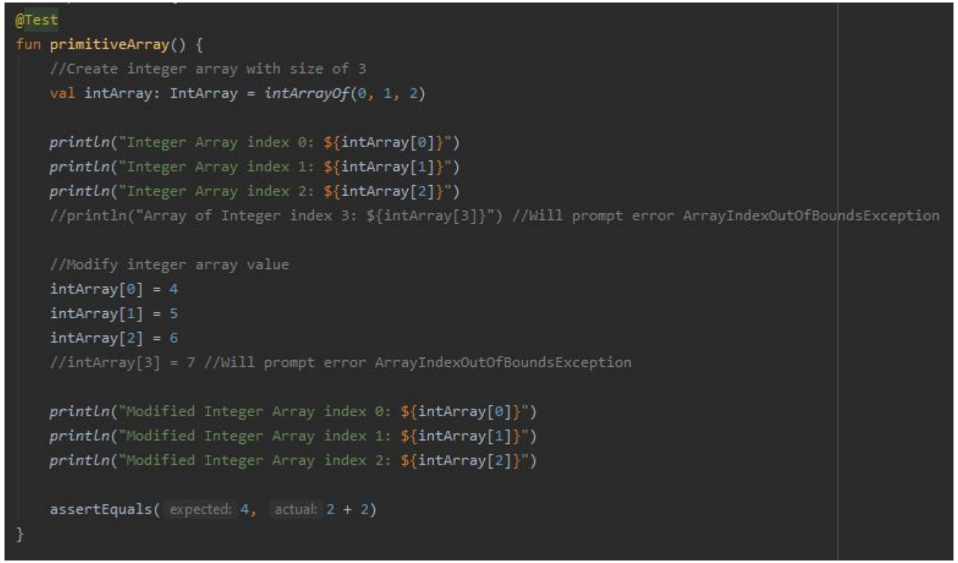
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Smile: 😄

Process finished with exit code 0
```

1.7.1 [Demo] Array and Collections

https://github.com/ferryyuwono/android-beginner/tree/main/1.07.01-Demo-ArrayCollections

File: ExampleUnitTest.kt

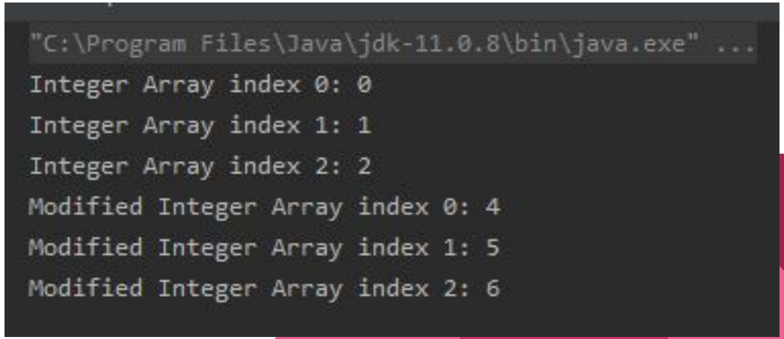
Array



```kotlin
@Test
fun primitiveArray() {
    //Create integer array with size of 3
    val intArray: IntArray = intArrayOf(0, 1, 2)

    println("Integer Array index 0: ${intArray[0]}")
    println("Integer Array index 1: ${intArray[1]}")
    println("Integer Array index 2: ${intArray[2]}")
    //println("Array of Integer index 3: ${intArray[3]}") //Will prompt error ArrayIndexOutOfBoundsException

    //Modify integer array value
    intArray[0] = 4
    intArray[1] = 5
    intArray[2] = 6
    //intArray[3] = 7 //Will prompt error ArrayIndexOutOfBoundsException

    println("Modified Integer Array index 0: ${intArray[0]}")
    println("Modified Integer Array index 1: ${intArray[1]}")
    println("Modified Integer Array index 2: ${intArray[2]}")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Array:

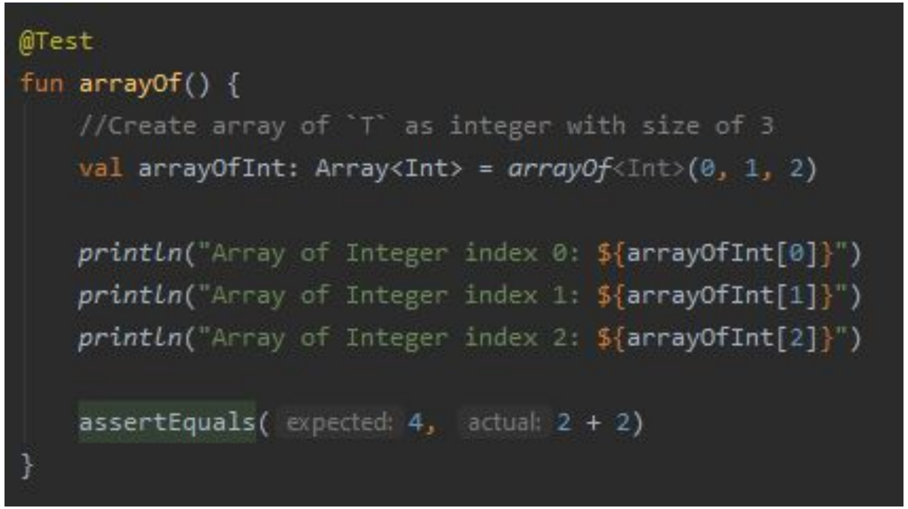
Data Type: IntArray, LongArray, etc
Assignment: intArrayOf(values separate by comma)



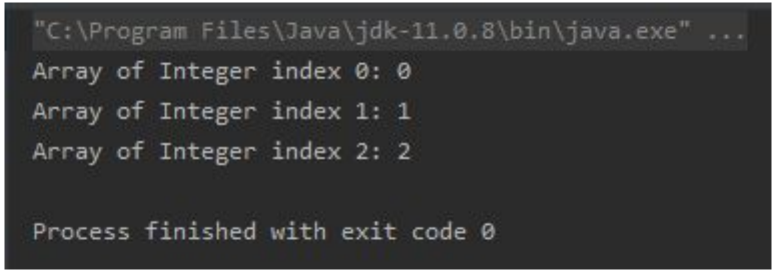
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Integer Array index 0: 0
Integer Array index 1: 1
Integer Array index 2: 2
Modified Integer Array index 0: 4
Modified Integer Array index 1: 5
Modified Integer Array index 2: 6
```

Array:

Data Type: Array<data type>
Assignment: arrayOf<data type>(values separate by comma)



```
@Test
fun arrayOf() {
    //Create array of `T` as integer with size of 3
    val arrayOfInt: Array<Int> = arrayOf<Int>(0, 1, 2)

    println("Array of Integer index 0: ${arrayOfInt[0]}")
    println("Array of Integer index 1: ${arrayOfInt[1]}")
    println("Array of Integer index 2: ${arrayOfInt[2]}")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

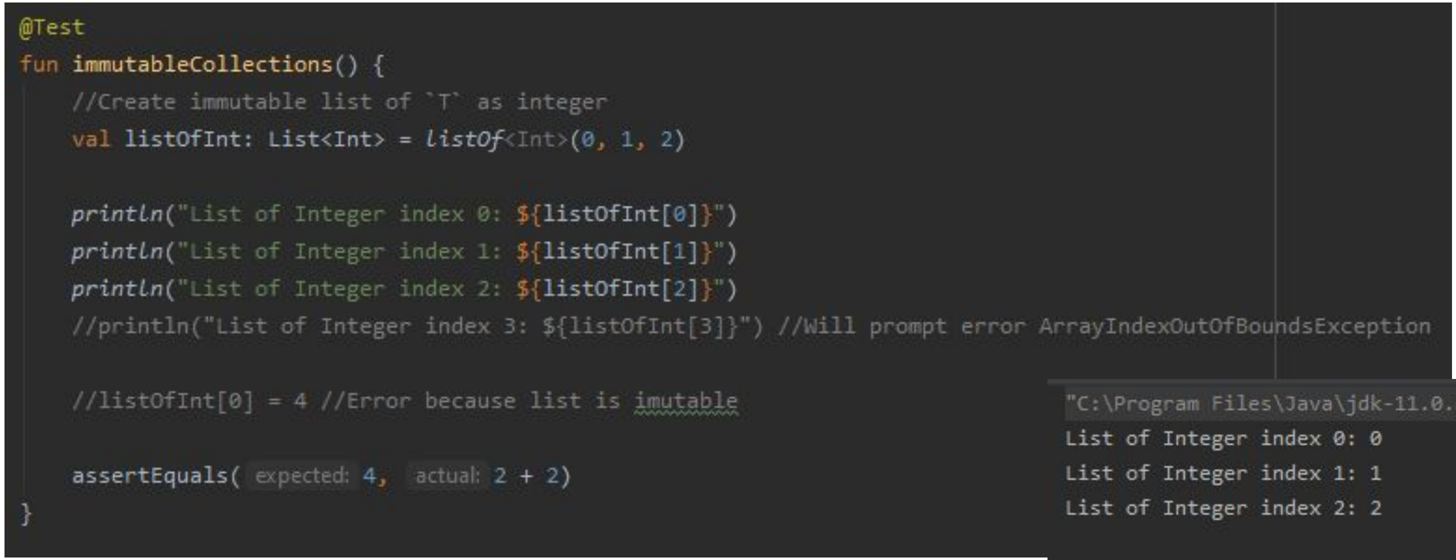


```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Array of Integer index 0: 0
Array of Integer index 1: 1
Array of Integer index 2: 2

Process finished with exit code 0
```

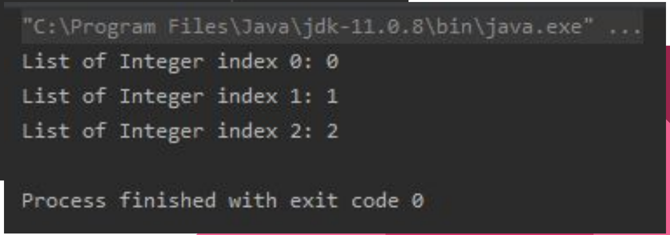
Collections

Collections:

Data Type: List<data type>
Assignment: listOf<data type>(values separate by comma)



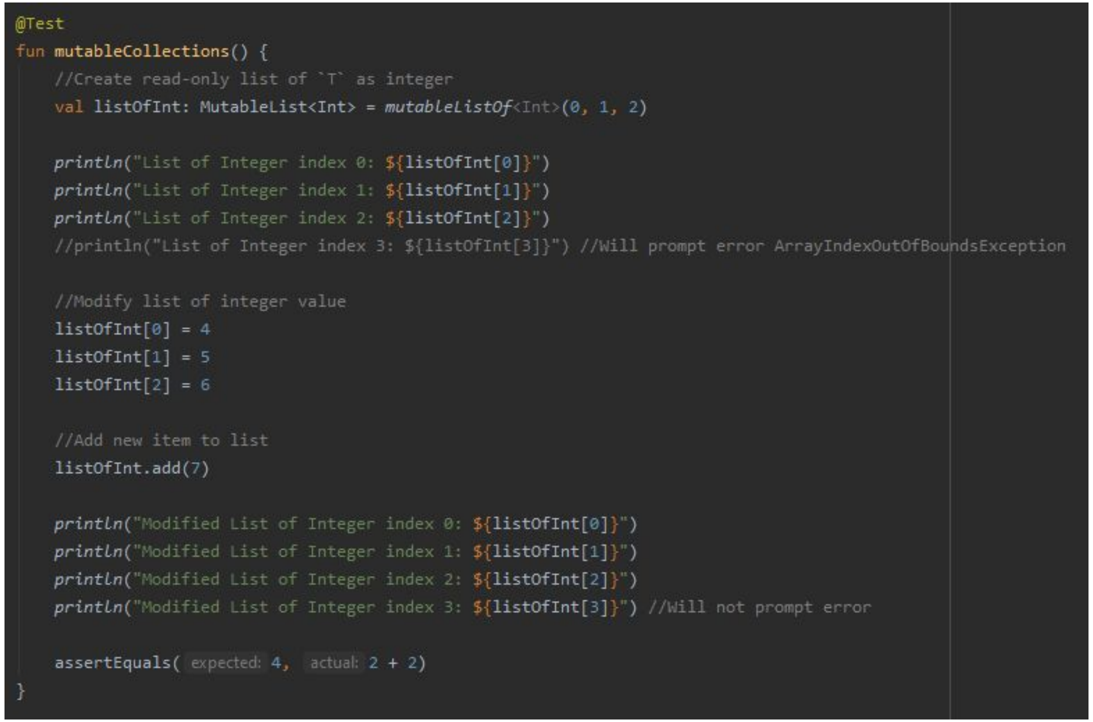
```kotlin
@Test
fun immutableCollections() {
    //Create immutable list of `T` as integer
    val listOfInt: List<Int> = listOf<Int>(0, 1, 2)

    println("List of Integer index 0: ${listOfInt[0]}")
    println("List of Integer index 1: ${listOfInt[1]}")
    println("List of Integer index 2: ${listOfInt[2]}")
    //println("List of Integer index 3: ${listOfInt[3]}") //Will prompt error ArrayIndexOutOfBoundsException

    //listOfInt[0] = 4 //Error because list is imutable

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



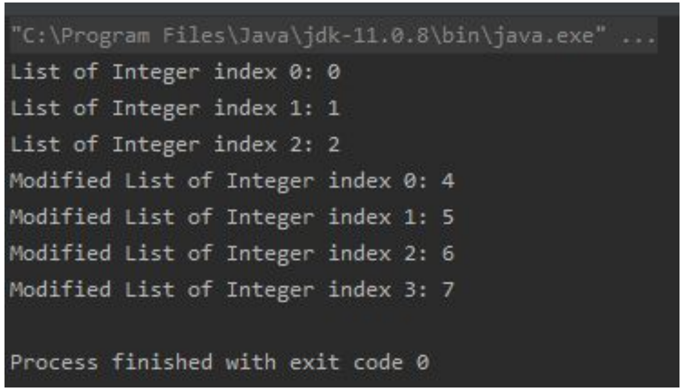
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
List of Integer index 0: 0
List of Integer index 1: 1
List of Integer index 2: 2

Process finished with exit code 0
```

Collections:

Data Type: MutableList<data type>
Assignment: mutableListOf<data type>(values separate by comma)
Add new data: listOfInt.add(new value)



```
@Test
fun mutableCollections() {
    //Create read-only list of `T` as integer
    val listOfInt: MutableList<Int> = mutableListOf<Int>(0, 1, 2)

    println("List of Integer index 0: ${listOfInt[0]}")
    println("List of Integer index 1: ${listOfInt[1]}")
    println("List of Integer index 2: ${listOfInt[2]}")
    //println("List of Integer index 3: ${listOfInt[3]}") //Will prompt error ArrayIndexOutOfBoundsException

    //Modify list of integer value
    listOfInt[0] = 4
    listOfInt[1] = 5
    listOfInt[2] = 6

    //Add new item to list
    listOfInt.add(7)

    println("Modified List of Integer index 0: ${listOfInt[0]}")
    println("Modified List of Integer index 1: ${listOfInt[1]}")
    println("Modified List of Integer index 2: ${listOfInt[2]}")
    println("Modified List of Integer index 3: ${listOfInt[3]}") //Will not prompt error

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
List of Integer index 0: 0
List of Integer index 1: 1
List of Integer index 2: 2
Modified List of Integer index 0: 4
Modified List of Integer index 1: 5
Modified List of Integer index 2: 6
Modified List of Integer index 3: 7

Process finished with exit code 0
```

1.7.2 [Exercise] Array and Collections

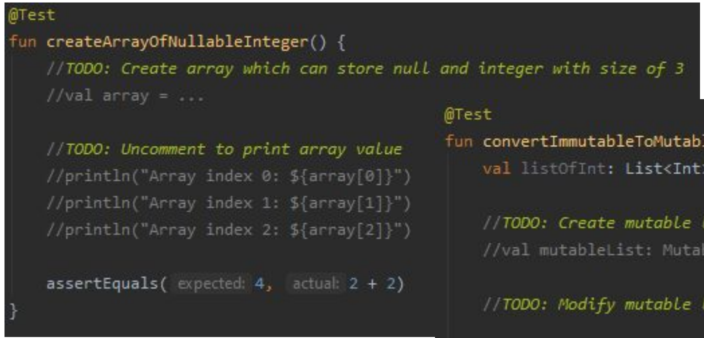
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.07.02-Exercise-ArrayCollections

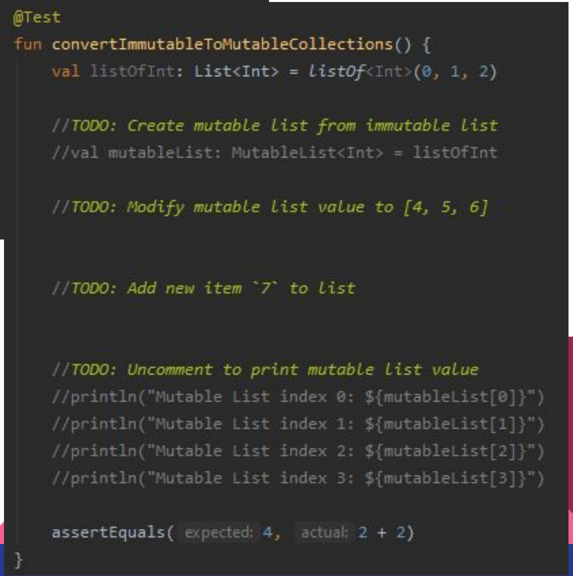
File

ExampleUnitTest.kt

- Array of nullable data type
- Convert immutable to mutable collection



```kotlin
@Test
fun createArrayOfNullableInteger() {
    //TODO: Create array which can store null and integer with size of 3
    //val array = ...

    //TODO: Uncomment to print array value
    //println("Array index 0: ${array[0]}")
    //println("Array index 1: ${array[1]}")
    //println("Array index 2: ${array[2]}")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```kotlin
@Test
fun convertImmutableToMutableCollections() {
    val listOfInt: List<Int> = listOf<Int>(0, 1, 2)

    //TODO: Create mutable list from immutable list
    //val mutableList: MutableList<Int> = listOfInt

    //TODO: Modify mutable list value to [4, 5, 6]


    //TODO: Add new item `7` to list


    //TODO: Uncomment to print mutable list value
    //println("Mutable List index 0: ${mutableList[0]}")
    //println("Mutable List index 1: ${mutableList[1]}")
    //println("Mutable List index 2: ${mutableList[2]}")
    //println("Mutable List index 3: ${mutableList[3]}")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

1.7.3 [Solution] Array and Collections

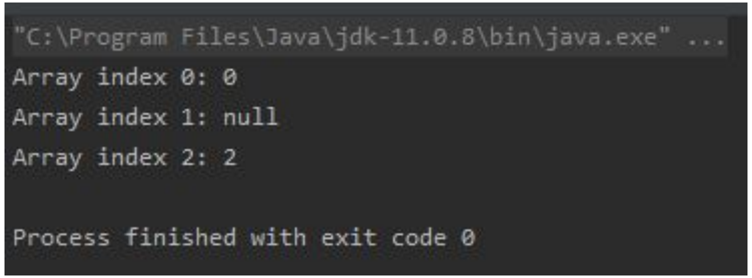
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.07.03-Solution-ArrayCollections

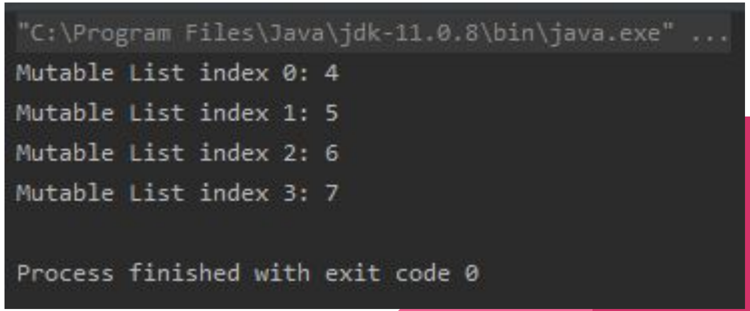
File

ExampleUnitTest.kt

- Array of nullable data type
- Convert immutable to mutable collection



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Array index 0: 0
Array index 1: null
Array index 2: 2

Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Mutable List index 0: 4
Mutable List index 1: 5
Mutable List index 2: 6
Mutable List index 3: 7

Process finished with exit code 0
```

1.8.1 [Demo] Conditions

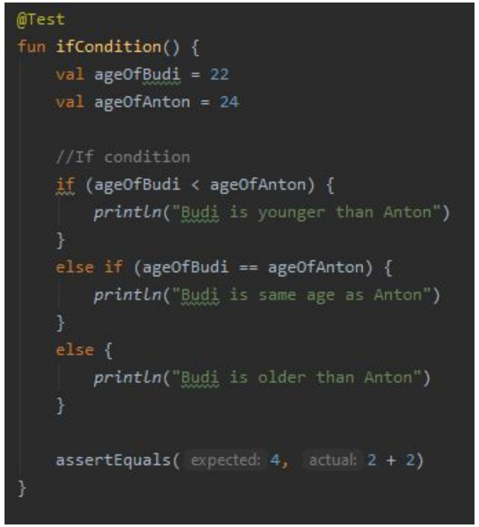
https://github.com/ferryyuwono/android-beginner/tree/main/1.08.01-Demo-Conditions

File: ExampleUnitTest.kt

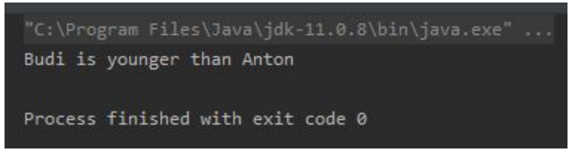
if condition

Format:

if (condition 1) {
    run this code if condition 1 is true
}
else if (condition 2) { //This **else if** is optional
    run this code if condition 2 is true
}
else { //This **else** is optional
    run this code if no condition is true
}

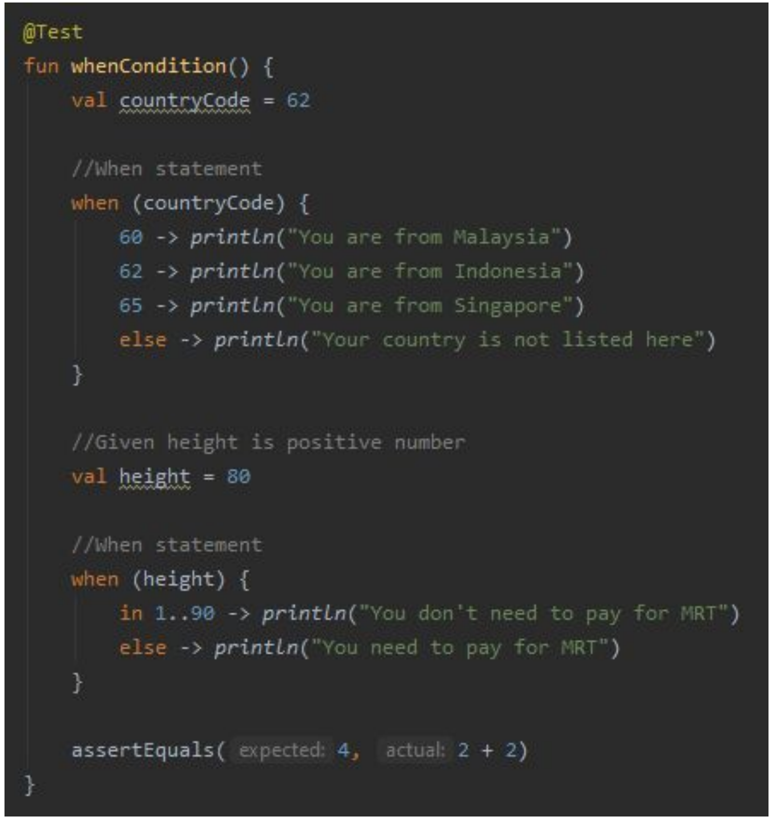


```
@Test
fun ifCondition() {
    val ageOfBudi = 22
    val ageOfAnton = 24

    //If condition
    if (ageOfBudi < ageOfAnton) {
        println("Budi is younger than Anton")
    }
    else if (ageOfBudi == ageOfAnton) {
        println("Budi is same age as Anton")
    }
    else {
        println("Budi is older than Anton")
    }

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Budi is younger than Anton

Process finished with exit code 0
```

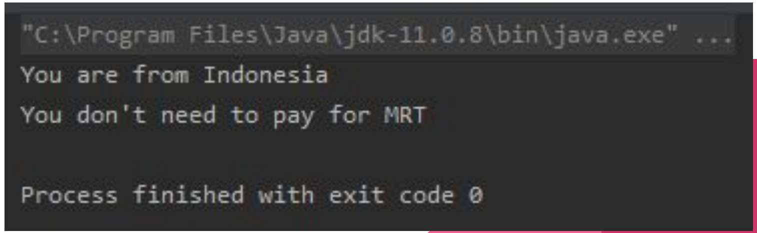
when condition



```
@Test
fun whenCondition() {
    val countryCode = 62

    //When statement
    when (countryCode) {
        60 -> println("You are from Malaysia")
        62 -> println("You are from Indonesia")
        65 -> println("You are from Singapore")
        else -> println("Your country is not listed here")
    }

    //Given height is positive number
    val height = 80

    //When statement
    when (height) {
        in 1..90 -> println("You don't need to pay for MRT")
        else -> println("You need to pay for MRT")
    }

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Format:

```
when (variable name) {
    value 1 -> {
        run this code when equals with value 1
    }
    value 2 -> {
        run this code when equals with value 2
    }
    else -> {
        run this code when no match value found
    }
}
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
You are from Indonesia
You don't need to pay for MRT

Process finished with exit code 0
```

1.8.2 [Exercise] Conditions

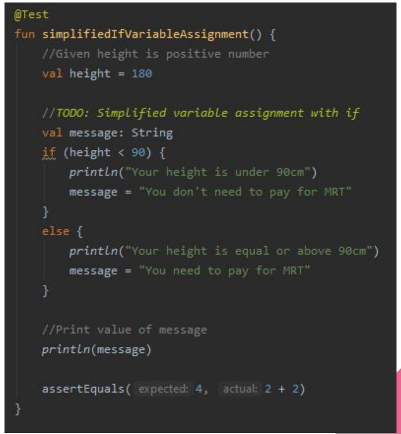
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.08.02-Exercise-Conditions

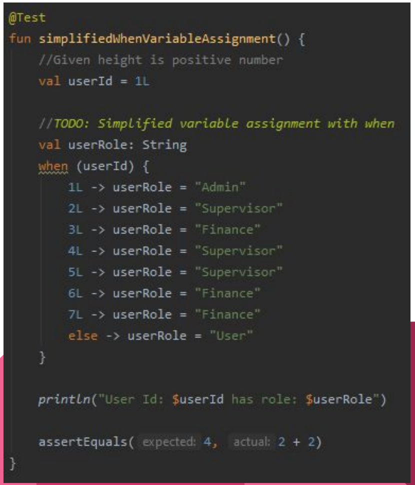
File

ExampleUnitTest.kt

- Variable assignment with condition
- Write simpler when condition



```kotlin
@Test
fun simplifiedIfVariableAssignment() {
    //Given height is positive number
    val height = 180

    //TODO: Simplified variable assignment with if
    val message: String
    if (height < 90) {
        println("Your height is under 90cm")
        message = "You don't need to pay for MRT"
    }
    else {
        println("Your height is equal or above 90cm")
        message = "You need to pay for MRT"
    }

    //Print value of message
    println(message)

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```kotlin
@Test
fun simplifiedWhenVariableAssignment() {
    //Given height is positive number
    val userId = 1L

    //TODO: Simplified variable assignment with when
    val userRole: String
    when (userId) {
        1L -> userRole = "Admin"
        2L -> userRole = "Supervisor"
        3L -> userRole = "Finance"
        4L -> userRole = "Supervisor"
        5L -> userRole = "Supervisor"
        6L -> userRole = "Finance"
        7L -> userRole = "Finance"
        else -> userRole = "User"
    }

    println("User Id: $userId has role: $userRole")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

1.8.3 [Solution] Conditions

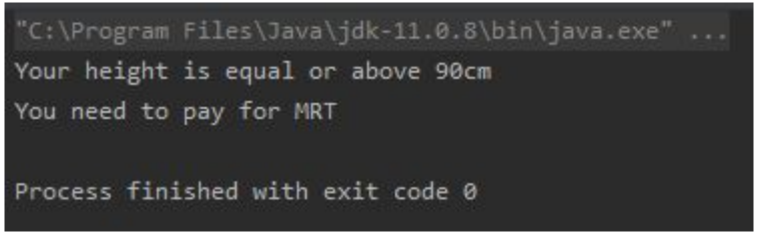
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.08.03-Solution-Conditions

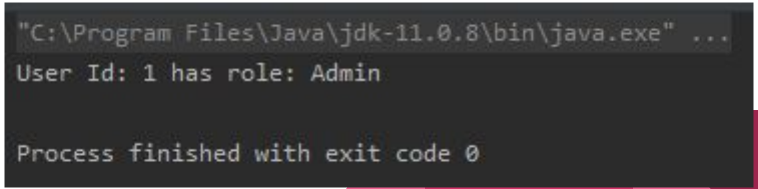
File

ExampleUnitTest.kt

- Variable assignment with condition
- Write simpler when condition



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Your height is equal or above 90cm
You need to pay for MRT

Process finished with exit code 0
```



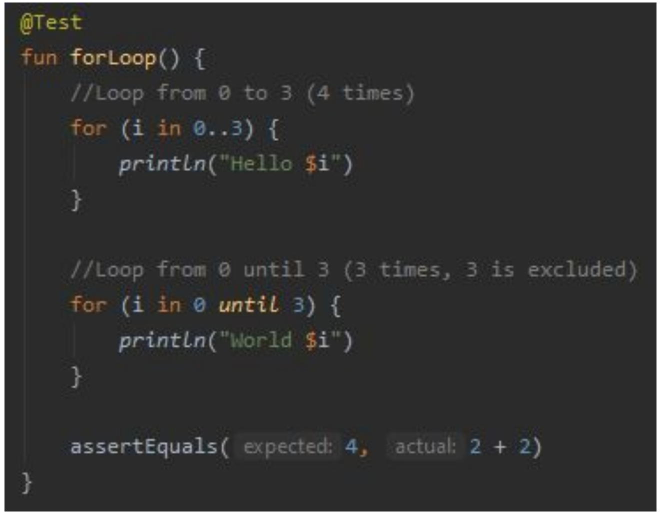
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
User Id: 1 has role: Admin

Process finished with exit code 0
```

1.9.1 [Demo] Loops

https://github.com/ferryyuwono/android-beginner/tree/main/1.09.01-Demo-Loops

File: ExampleUnitTest.kt

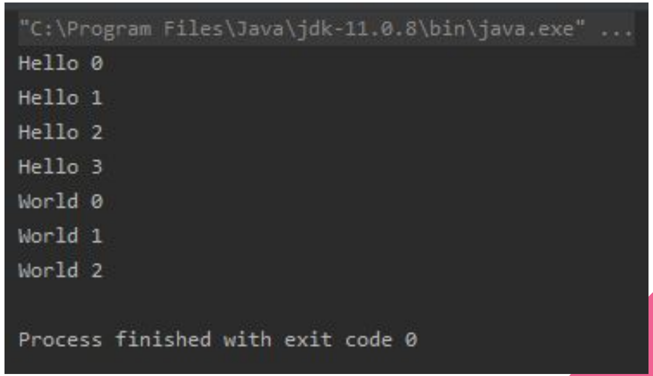
For loops



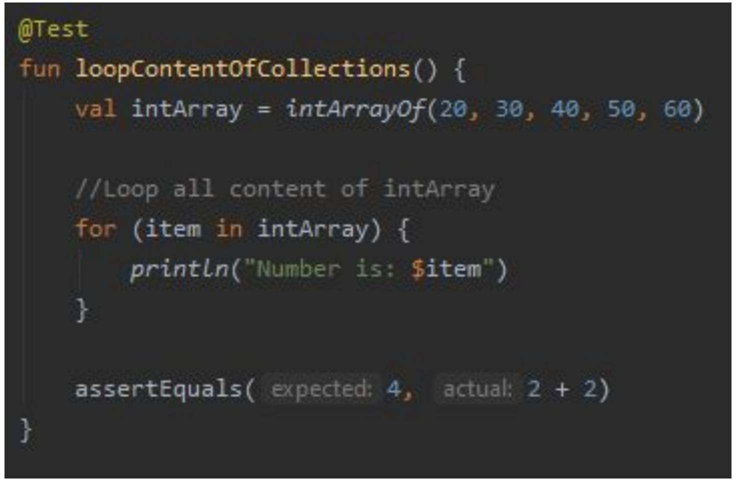
```
@Test
fun forLoop() {
    //Loop from 0 to 3 (4 times)
    for (i in 0..3) {
        println("Hello $i")
    }

    //Loop from 0 until 3 (3 times, 3 is excluded)
    for (i in 0 until 3) {
        println("World $i")
    }

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Format:

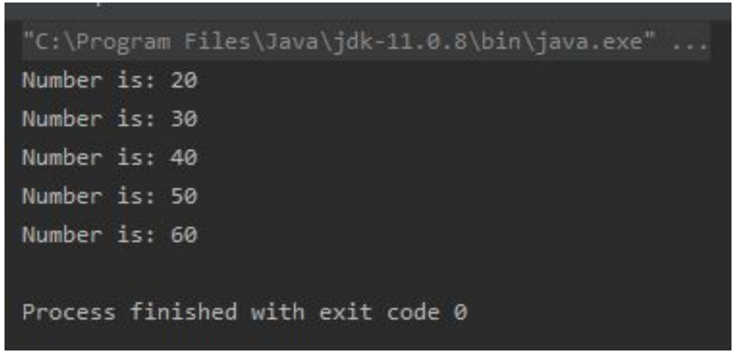
for (loop variable in loop range) {
    run this code until loop is done
}



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Hello 0
Hello 1
Hello 2
Hello 3
World 0
World 1
World 2

Process finished with exit code 0
```



```
@Test
fun loopContentOfCollections() {
    val intArray = intArrayOf(20, 30, 40, 50, 60)

    //Loop all content of intArray
    for (item in intArray) {
        println("Number is: $item")
    }

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Number is: 20
Number is: 30
Number is: 40
Number is: 50
Number is: 60

Process finished with exit code 0
```

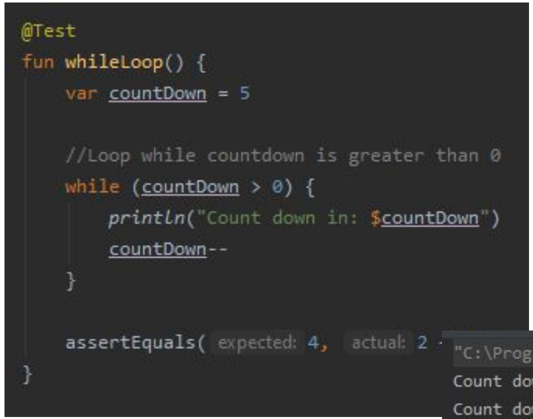
Format:

for (loop variable in collection) {
    run this code until each value of collection is looped
}

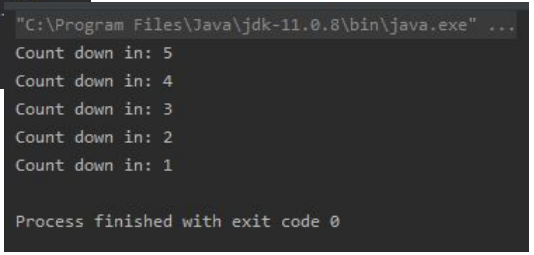
While Loops

Format:

while (while condition) {
    run this code while condition is true
}



```
@Test
fun whileLoop() {
    var countDown = 5

    //Loop while countdown is greater than 0
    while (countDown > 0) {
        println("Count down in: $countDown")
        countDown--
    }

    assertEquals( expected: 4,  actual: 2 -
}
```

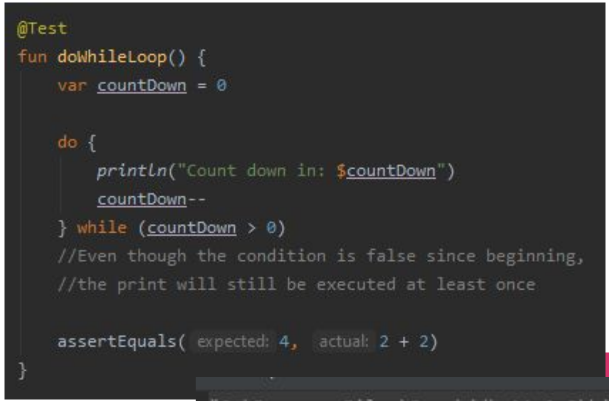


```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Count down in: 5
Count down in: 4
Count down in: 3
Count down in: 2
Count down in: 1

Process finished with exit code 0
```

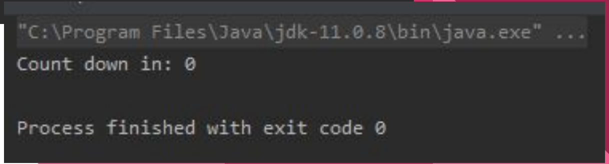
Do While Loops

Format:

do {
    run this code at least once
    then check while condition
} while (while condition)



```
@Test
fun doWhileLoop() {
    var countDown = 0

    do {
        println("Count down in: $countDown")
        countDown--
    } while (countDown > 0)
    //Even though the condition is false since beginning,
    //the print will still be executed at least once

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Count down in: 0

Process finished with exit code 0
```

1.9.2 [Exercise] Loops

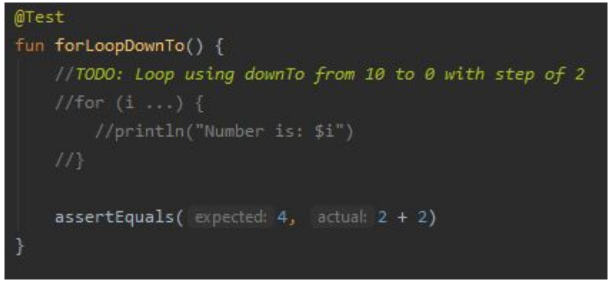
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.09.02-Exercise-Loops

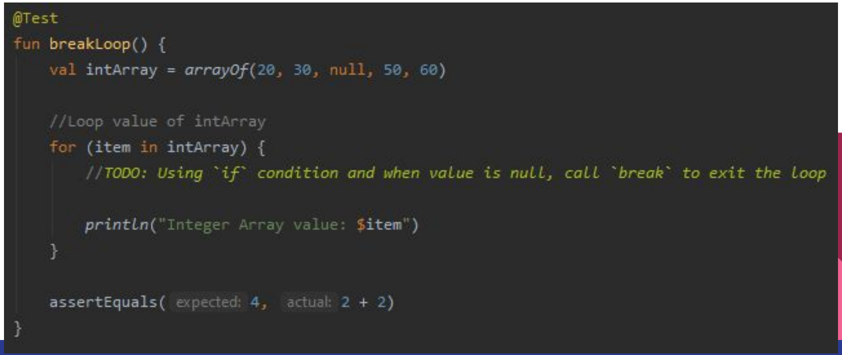
File

ExampleUnitTest.kt

- For loop using downTo and step
- Loop collection with index
- Break loop
- Skip data in loop



```kotlin
@Test
fun forLoopDownTo() {
    //TODO: Loop using downTo from 10 to 0 with step of 2
    //for (i ...) {
        //println("Number is: $i")
    //}

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```kotlin
@Test
fun breakLoop() {
    val intArray = arrayOf(20, 30, null, 50, 60)

    //Loop value of intArray
    for (item in intArray) {
        //TODO: Using `if` condition and when value is null, call `break` to exit the loop

        println("Integer Array value: $item")
    }

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

1.9.3 [Solution] Loops

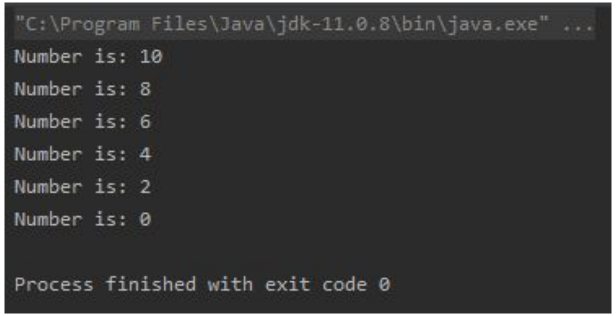
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.09.03-Solution-Loops

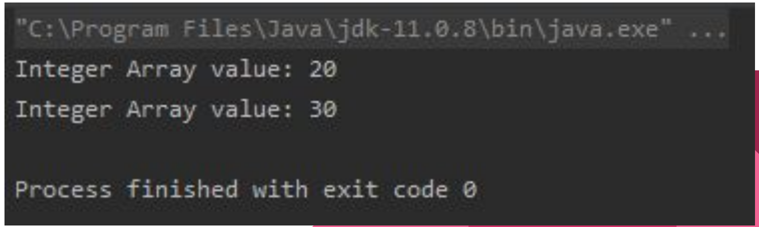
File

ExampleUnitTest.kt

- For loop using downTo and step
- Loop collection with index
- Break loop
- Skip data in loop



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Number is: 10
Number is: 8
Number is: 6
Number is: 4
Number is: 2
Number is: 0

Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Integer Array value: 20
Integer Array value: 30

Process finished with exit code 0
```

1.10.1 [Demo] Functions

https://github.com/ferryyuwono/android-beginner/tree/main/1.10.01-Demo-Functions

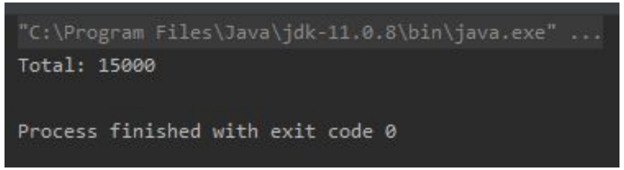
File: ExampleUnitTest.kt

Declare Function

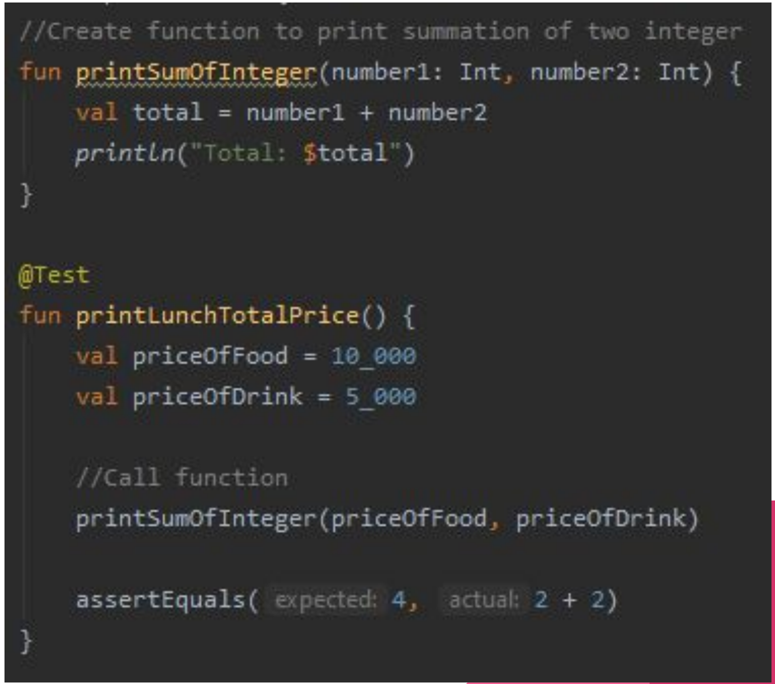
Format:

fun function name(parameter) {
    code inside function
}

function name: lower camelCase
parameter: separated by comma (optional)

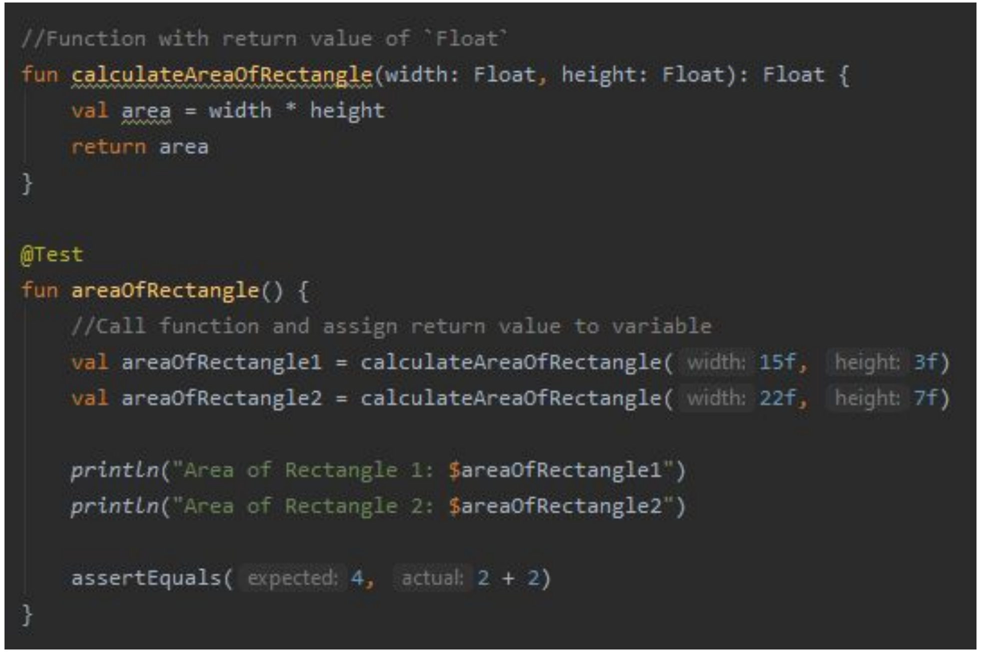


```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Total: 15000

Process finished with exit code 0
```



```kotlin
//Create function to print summation of two integer
fun printSumOfInteger(number1: Int, number2: Int) {
    val total = number1 + number2
    println("Total: $total")
}

@Test
fun printLunchTotalPrice() {
    val priceOfFood = 10_000
    val priceOfDrink = 5_000

    //Call function
    printSumOfInteger(priceOfFood, priceOfDrink)

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Function with return value



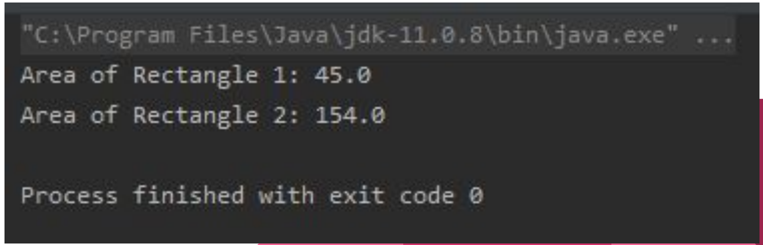
```
//Function with return value of `Float`
fun calculateAreaOfRectangle(width: Float, height: Float): Float {
    val area = width * height
    return area
}

@Test
fun areaOfRectangle() {
    //Call function and assign return value to variable
    val areaOfRectangle1 = calculateAreaOfRectangle( width: 15f,  height: 3f)
    val areaOfRectangle2 = calculateAreaOfRectangle( width: 22f,  height: 7f)

    println("Area of Rectangle 1: $areaOfRectangle1")
    println("Area of Rectangle 2: $areaOfRectangle2")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Format:

fun function name(parameter): data type {
    code inside function

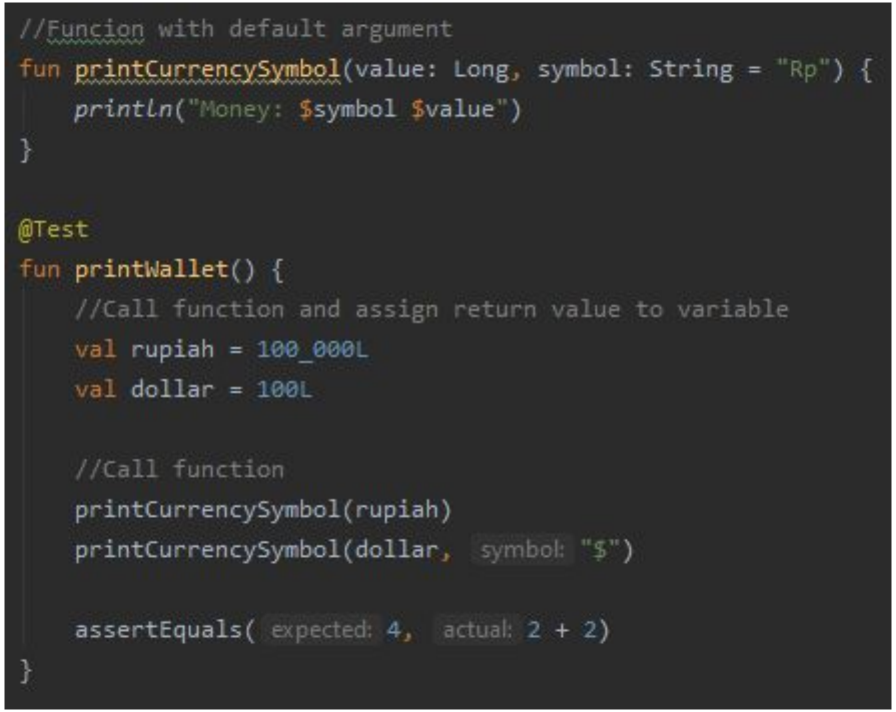
    return data type
}

function name: lower camelCase
parameter: separated by comma (optional)



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Area of Rectangle 1: 45.0
Area of Rectangle 2: 154.0

Process finished with exit code 0
```

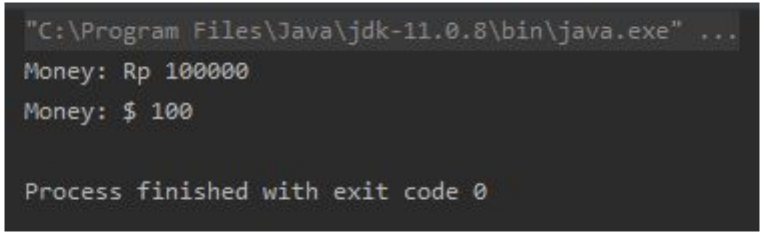
Function with default argument



```kotlin
//Funcion with default argument
fun printCurrencySymbol(value: Long, symbol: String = "Rp") {
    println("Money: $symbol $value")
}

@Test
fun printWallet() {
    //Call function and assign return value to variable
    val rupiah = 100_000L
    val dollar = 100L

    //Call function
    printCurrencySymbol(rupiah)
    printCurrencySymbol(dollar,  symbol: "$")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Format:

fun function name(parameter = default value) {
    code inside function
}



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Money: Rp 100000
Money: $ 100

Process finished with exit code 0
```

1.10.2 [Exercise] Functions

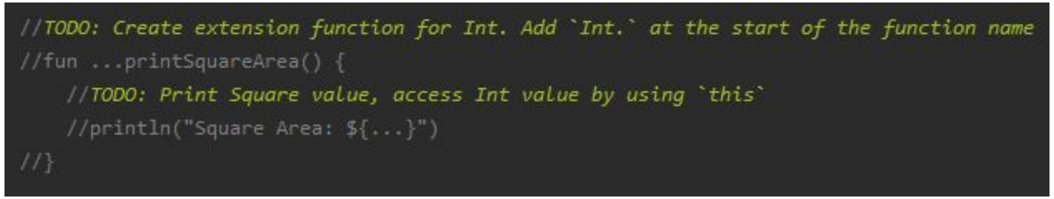
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.10.02-Exercise-Functions

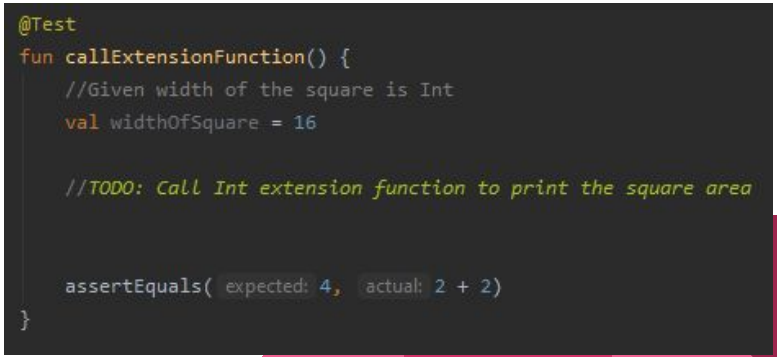
File

ExampleUnitTest.kt

- Write single expression function
- Using named argument in parameter
- Extension function



```kotlin
//TODO: Create extension function for Int. Add `Int.` at the start of the function name
//fun ...printSquareArea() {
    //TODO: Print Square value, access Int value by using `this`
    //println("Square Area: ${...}")
//}
```



```kotlin
@Test
fun callExtensionFunction() {
    //Given width of the square is Int
    val widthOfSquare = 16

    //TODO: Call Int extension function to print the square area


    assertEquals( expected: 4,  actual: 2 + 2)
}
```

1.10.3 [Solution] Functions

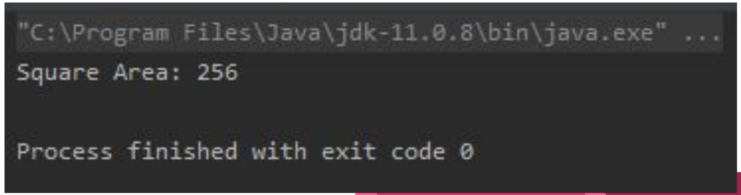
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.10.03-Solution-Functions

File

ExampleUnitTest.kt

- Write single expression function
- Using named argument in parameter
- Extension function



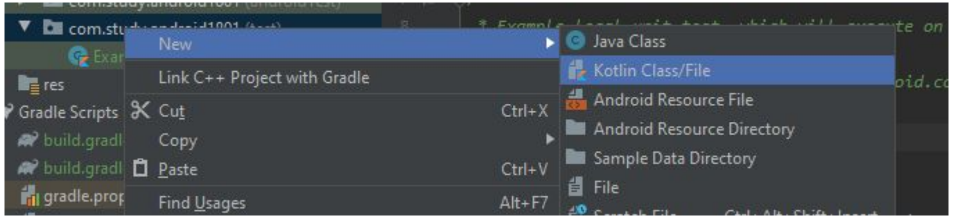
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Square Area: 256

Process finished with exit code 0
```

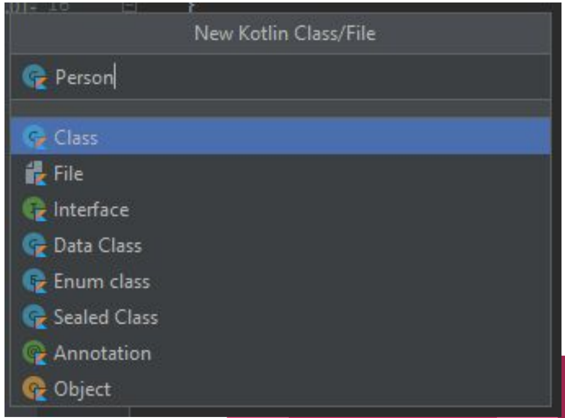
1.11.1 [Demo] Class and Data Class

https://github.com/ferryyuwono/android-beginner/tree/main/1.11.01-Demo-ClassDataClass

File: ExampleUnitTest.kt

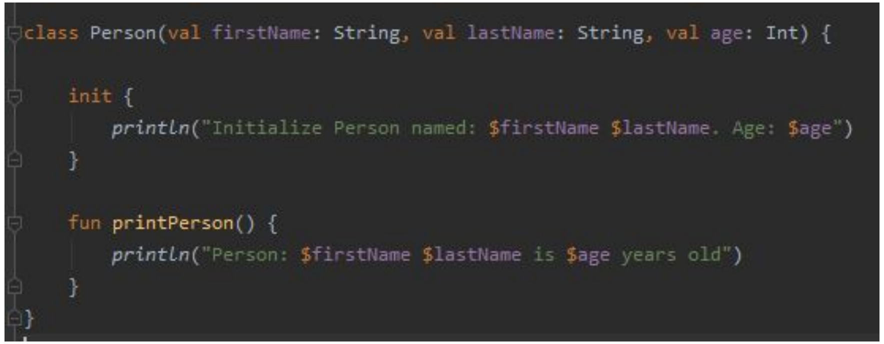
Create new Kotlin Class



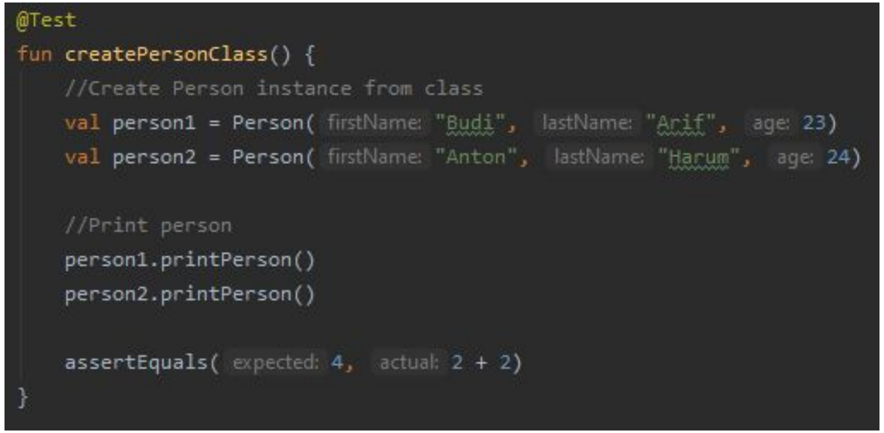


Class File

Person.kt



```
class Person(val firstName: String, val lastName: String, val age: Int) {

    init {
        println("Initialize Person named: $firstName $lastName. Age: $age")
    }

    fun printPerson() {
        println("Person: $firstName $lastName is $age years old")
    }
}
```

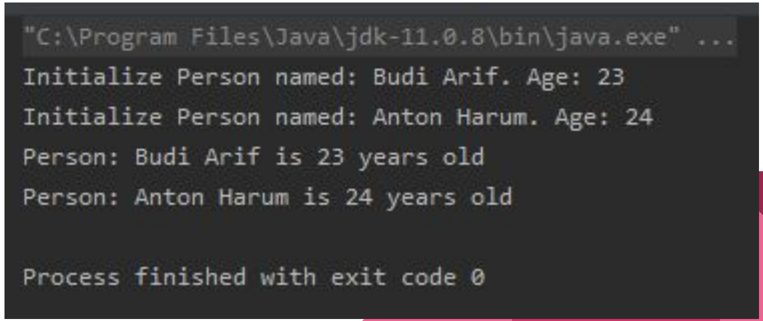


```
@Test
fun createPersonClass() {
    //Create Person instance from class
    val person1 = Person( firstName: "Budi",  lastName: "Arif",  age: 23)
    val person2 = Person( firstName: "Anton",  lastName: "Harum",  age: 24)

    //Print person
    person1.printPerson()
    person2.printPerson()

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

Format:

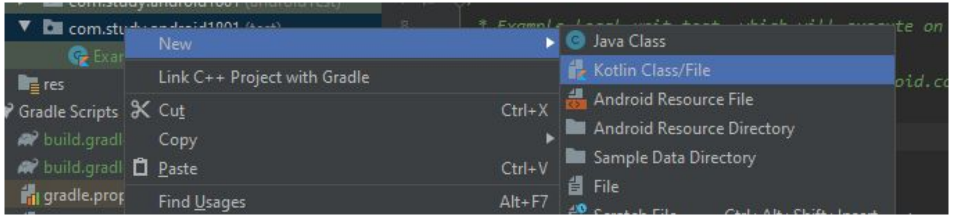
class class name(parameter) {
  init { }
}

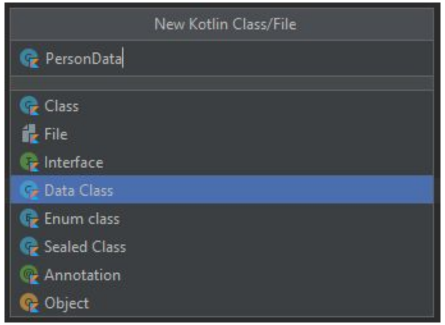
class name: PascalCase (start with capital letter)
parameter: separated by comma (optional)
init { }: constructor function which will be called when object of the class is created (optional)



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Initialize Person named: Budi Arif. Age: 23
Initialize Person named: Anton Harum. Age: 24
Person: Budi Arif is 23 years old
Person: Anton Harum is 24 years old

Process finished with exit code 0
```

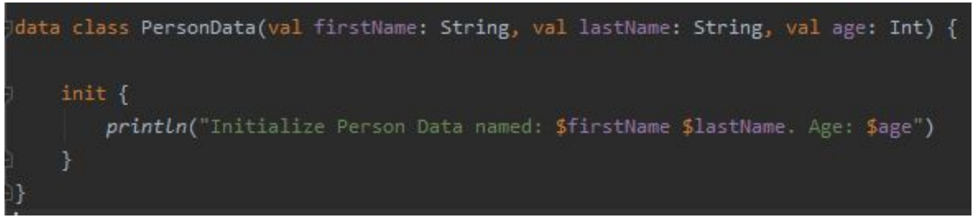
Create Data Class





Data Class File

PersonData.kt

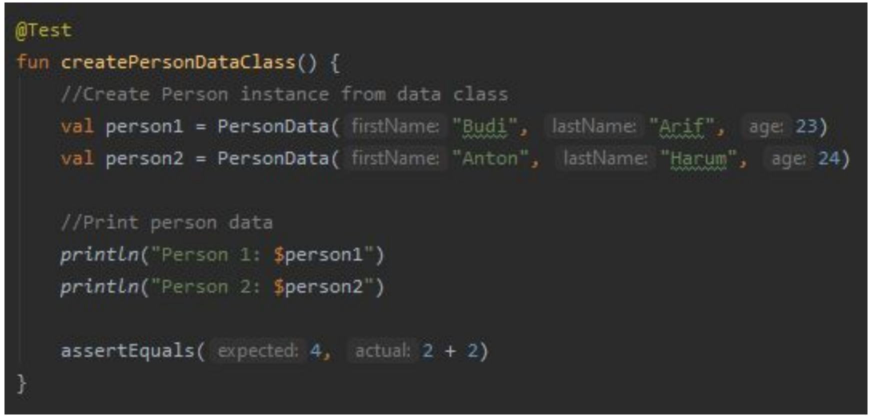


```
data class PersonData(val firstName: String, val lastName: String, val age: Int) {

    init {
        println("Initialize Person Data named: $firstName $lastName. Age: $age")
    }
}
```

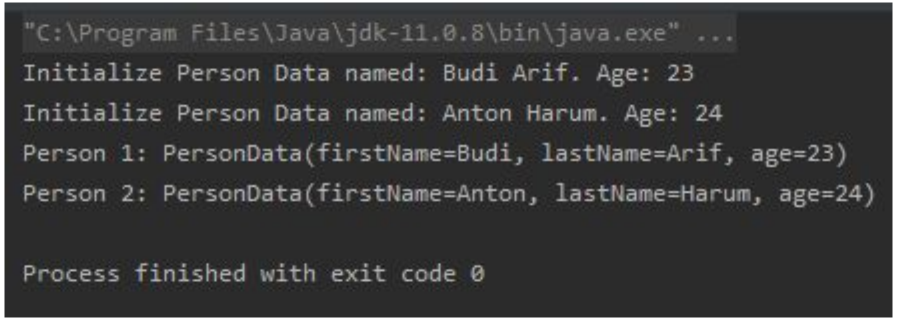
Format:

data class class name(parameter) {
  init { }
}

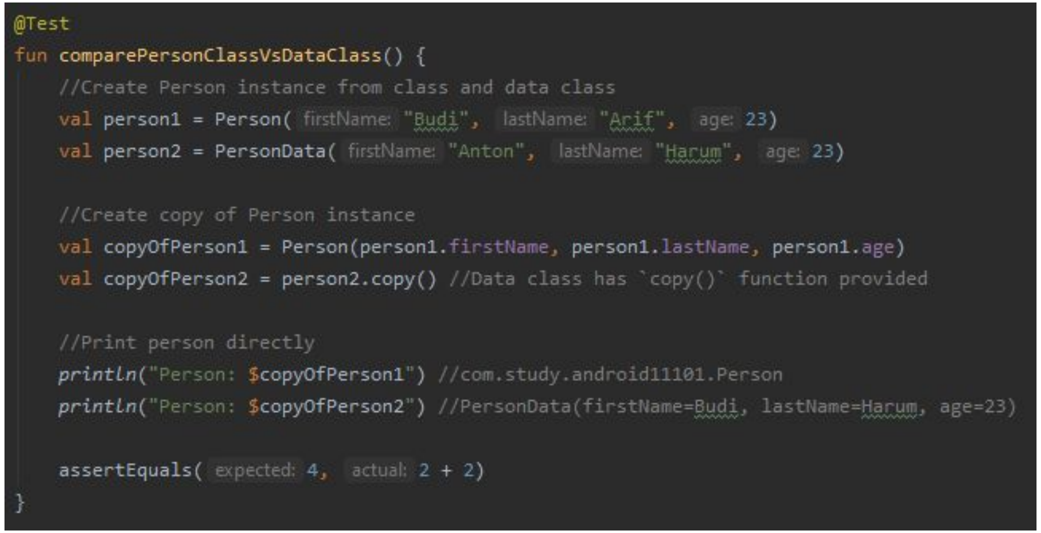
class name: PascalCase (start with capital letter)
parameter: at least 1 parameter (**mandatory**)
init { }: constructor function which will be called when object of the data class is created (optional)



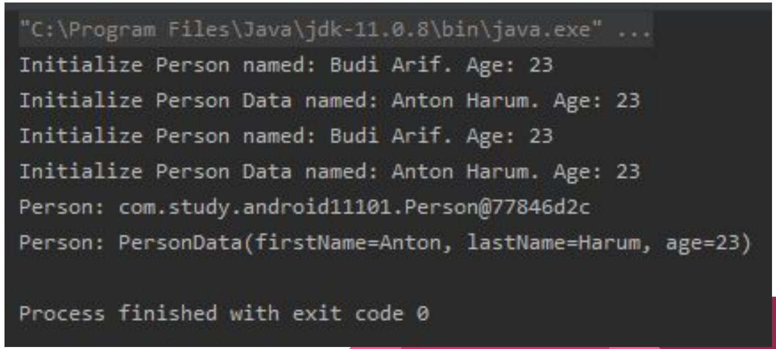
```kotlin
@Test
fun createPersonDataClass() {
    //Create Person instance from data class
    val person1 = PersonData( firstName: "Budi",  lastName: "Arif",  age: 23)
    val person2 = PersonData( firstName: "Anton",  lastName: "Harum",  age: 24)

    //Print person data
    println("Person 1: $person1")
    println("Person 2: $person2")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Initialize Person Data named: Budi Arif. Age: 23
Initialize Person Data named: Anton Harum. Age: 24
Person 1: PersonData(firstName=Budi, lastName=Arif, age=23)
Person 2: PersonData(firstName=Anton, lastName=Harum, age=24)

Process finished with exit code 0
```



```kotlin
@Test
fun comparePersonClassVsDataClass() {
    //Create Person instance from class and data class
    val person1 = Person( firstName: "Budi",  lastName: "Arif",  age: 23)
    val person2 = PersonData( firstName: "Anton",  lastName: "Harum",  age: 23)

    //Create copy of Person instance
    val copyOfPerson1 = Person(person1.firstName, person1.lastName, person1.age)
    val copyOfPerson2 = person2.copy() //Data class has `copy()` function provided

    //Print person directly
    println("Person: $copyOfPerson1") //com.study.android11101.Person
    println("Person: $copyOfPerson2") //PersonData(firstName=Budi, lastName=Harum, age=23)

    assertEquals( expected: 4,  actual: 2 + 2)
}
```



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Initialize Person named: Budi Arif. Age: 23
Initialize Person Data named: Anton Harum. Age: 23
Initialize Person named: Budi Arif. Age: 23
Initialize Person Data named: Anton Harum. Age: 23
Person: com.study.android11101.Person@77846d2c
Person: PersonData(firstName=Anton, lastName=Harum, age=23)

Process finished with exit code 0
```

1.11.2 [Exercise] Class And Data Class

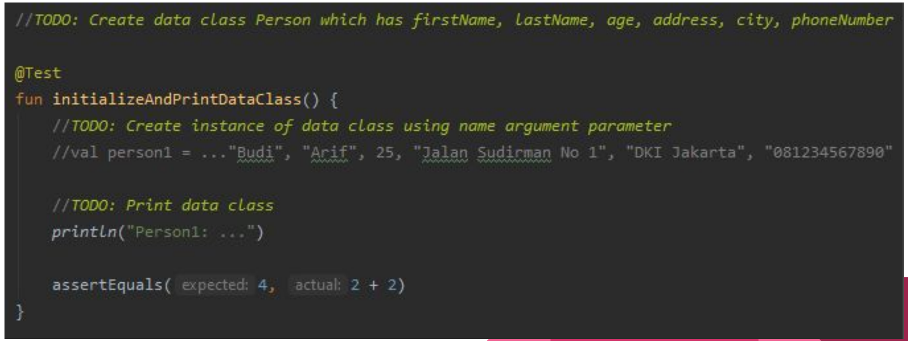
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.11.02-Exercise-ClassDataClass

File

ExampleUnitTest.kt

- Create instance of class using named argument
- Companion object



```kotlin
//TODO: Create data class Person which has firstName, lastName, age, address, city, phoneNumber

@Test
fun initializeAndPrintDataClass() {
    //TODO: Create instance of data class using name argument parameter
    //val person1 = ..."Budi", "Arif", 25, "Jalan Sudirman No 1", "DKI Jakarta", "081234567890"

    //TODO: Print data class
    println("Person1: ...")

    assertEquals( expected: 4,  actual: 2 + 2)
}
```

1.11.3 [Solution] Class And Data Class

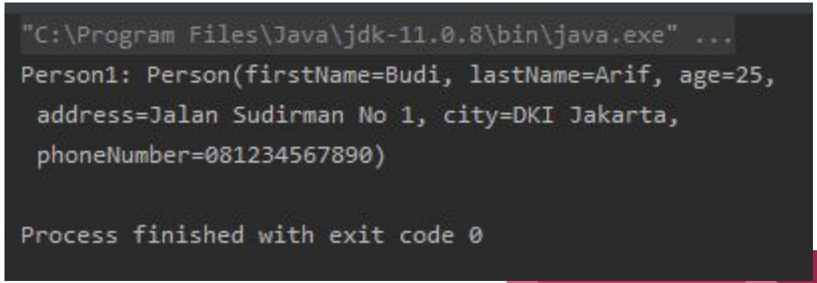
Repository

https://github.com/ferryyuwono/android-beginner/tree/main/1.11.03-Solution-ClassDataClass

File

ExampleUnitTest.kt

- Create instance of class using named argument
- Companion object



```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" ...
Person1: Person(firstName=Budi, lastName=Arif, age=25,
 address=Jalan Sudirman No 1, city=DKI Jakarta,
 phoneNumber=081234567890)

Process finished with exit code 0
```

End of Module 1:
Good Job! See you in the next module