

# Recording in Progress

This class is being recorded

Please turn off your video and/or video if you do not wish to be recorded

# **CMSC436: Programming Handheld Systems**

# **Android Development Environment**

# The Android Platform

A multi-layered software stack for building and running mobile applications

# The Android Development Environment

Starts with knowledge of the Android platform

Your workbench for writing Android applications

See:

<https://developer.android.com/studio/intro/>

# Today's Topics

Downloading Android SDK

Using the Android Studio IDE

Using the Android emulator

Debugging Android applications

Other tools

# Prerequisites

## Supported Operating Systems:

Microsoft Windows 8/10 (64-bit)

Mac OS X 10.14 (Mojave) or higher

Any 64-bit Linux that supports Gnome, KDE, Unity DE

# General Prerequisites

8GB RAM min

8GB memory for Android SDK, emulator system images, and caches

1280 x 800 min screen resolution



# Getting Started

Download & install Android Studio

See: <https://developer.android.com/studio/>

# Android Studio

Android platform

Android Studio IDE

Key development tools

System image for emulator

HelloWorld



# The Android Emulator

Runs virtual devices

We'll generally be using a Pixel 5 with API level 33

Android Studio interface showing the MainActivity.kt file and the Device Manager panel.

```
examples - MainActivity.kt [MapLocation.app.main]
examples \ MapLocation \ app \ src \ main \ java \ course \ examples \ maplocation \ MainActivity.kt
39
40
41 /**
42  * This helper method is called when user clicks the
43  * Show Map button and will start an activity to display
44  * a user specified location.
45  */
46 private fun processClick() {
47     try {
48         // Process text for network transmission
49         val address =
50             binding.location.text
51                 .toString()
52                 .replace(" ", "+")
53
54         // Create Intent object for starting Google Maps application
55         val geoIntent = Intent(
56             Intent.ACTION_VIEW,
57             Uri.parse("geo:0,0?q=$address")
58         )
59
60         // Call resolveIntent to allow the user to possibly choose
61         // their preferred application to display this geo location.
62         // If the system only has one application that can perform
63         // this request, that application will be started with no user
64         // interaction.
65         geoIntent.resolveActivity(packageManager)?.let { it: ComponentName?
66             // An activity exists for this action intent, so start
67             // that activity.
68             startActivity(geoIntent)
69         }
70     } catch (e: Exception) {
71         // Log any error messages to LogCat using Log.e()
72         Log.e(TAG, e.toString())
73     }
74 }
75
76 override fun onStart() {
77     super.onStart()
78     Log.i(TAG, msg= "The activity is visible and about to be started.")
79 }
80
81 override fun onRestart() {
82     super.onRestart()
83     Log.i(TAG, msg= "The activity is visible and about to be restarted.")
84 }
85
86 override fun onResume() {
87     super.onResume()
88     Log.i(TAG, msg= "The activity is visible and has focus (it is now \"resumed\")")
89 }
90
91 override fun onPause() {
92     super.onPause()
93     Log.i(
94         TAG,
95         msg= "Another activity is taking focus (this activity is about to be \"paused\")"
96     )
97 }
```

Device Manager:

Device	API	Size on Disk	Actions
Pixel 2 API 31	31	15 GB	▶ 🗑️ ✎
Pixel 3 API 31	31	9.7 GB	▶ 🗑️ ✎
Pixel C API 31	31	9.4 GB	▶ 🗑️ ✎

Two yellow arrows point to the 'Create device' button and the 'Pixel 2 API 31' device entry.

examples - MainActivity.kt [MapLocation.app.main]

```
examples \ MapLocation \ app \ src \ main \ java \ course \ examples \ maplocation \ MainActivity.kt \ processClick
```

Project

- examples
- MapLocation
- app
- src
- main
- java
- course
- examples
- maplocation
- MainActivity
- processClick

Device Manager

Device	API	Size on Disk	Actions
Pixel 2 API 30	Android 11.0 Google APIs   x86	30	9.0 GB
Pixel 2 API 31	Android 12.0 Google Play   x86_64	31	15 GB
Pixel 3 API 31	Android 12.0 Google Play   x86_64	31	9.7 GB
Pixel 3 API 31	Android 12.0 Google Play   x86_64	31	9.4 GB

Virtual Device Configuration

### Select Hardware

Choose a device definition

Category	Name	Play Store	Size	Resolution	Density
TV	Resizable		6.0"	1080x...	420dpi
Phone	Pixel XL		5.5"	1440x...	560dpi
Wear OS	Pixel 5		6.0"	1080x...	440dpi
Tablet	Pixel 4a		5.8"	1080x...	440dpi
Automotive	Pixel 4 XL		6.3"	1440x...	560dpi
	Pixel 4		5.7"	1080x...	440dpi
	Pixel 3a XL		6.0"	1080x...	400dpi
	Pixel 3a		5.6"	1080x...	440dpi
	Pixel 3 XL		6.3"	1440x...	560dpi
	<b>Pixel 3</b>		<b>5.46"</b>	<b>1080x...</b>	<b>440dpi</b>
	Pixel 2 XL		5.99"	1440x...	560dpi

Pixel 3

Size: large  
Ratio: long  
Density: 440dpi

New Hardware Profile Import Hardware Profiles Clone Device...

Cancel Previous Next Finish

# The Android Emulator

## Pros

Doesn't require an actual phone

Hardware is reconfigurable

Changes are non-destructive

# The Android Emulator

## Cons

Slower than an actual device

Some features unavailable

e.g., no support for NFC, SD card insert/eject, etc.

Performance / user experience can be misleading



# Advanced Features

Can emulate many different device/user characteristics, such as:

- Network speed/latencies

- Battery power

- Location coordinates

- Sensor readings

# Advanced Features

Ex: Change network speeds

Extended controls - Pixel\_2\_API\_30-5554

Location	Network type	Signal strength
Displays	EDGE	Moderate
Cellular	Voice status	Data status
Battery	Home	Home
Camera		
Phone		
Directional pad		
Microphone		
Fingerprint		
Virtual sensors		
Bug report		
Snapshots		
Record and Playback		
Settings		
Help		



# Advanced Features

Ex: Emulate incoming phone calls & SMS messages

Extended controls - Pixel\_2\_API\_30:5554

- Location
- Displays
- Cellular
- Battery
- Camera
- Phone
- Directional pad
- Microphone
- Fingerprint
- Virtual sensors
- Bug report
- Snapshots
- Record and Playback
- Settings
- Help

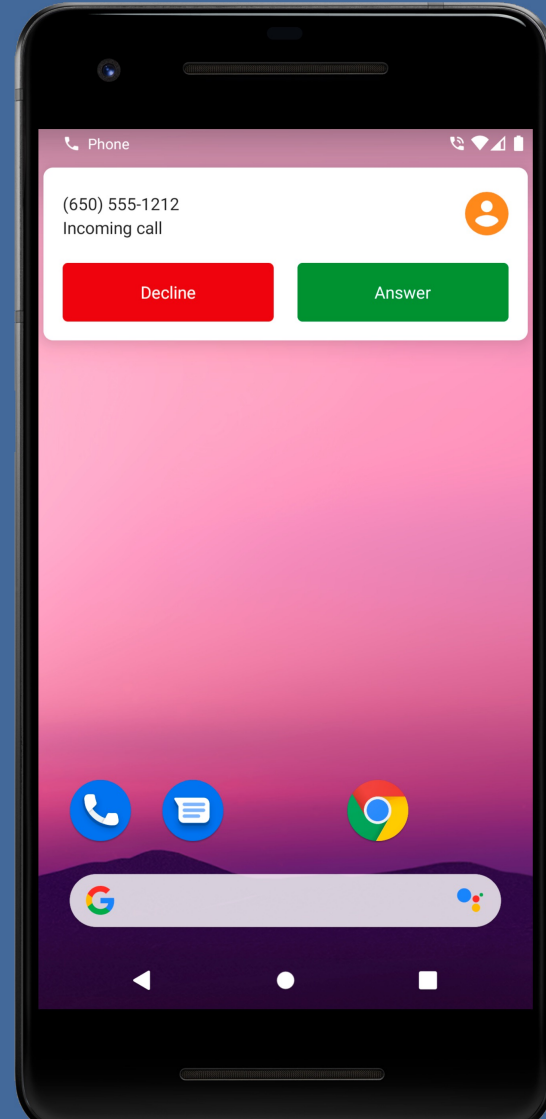
From (650) 555-1212

HOLD CALL    END CALL

SMS message

Yum! Pie à la Android mode!

SEND MESSAGE



- Power
- Volume
- Rotation
- Camera
- Search
- Navigation
- App Drawer
- More





# Advanced Features

Many more options

See:

<https://developer.android.com/studio/run/emulator.html>

# Debugger

Tool for examining the internal state of a running application



TheAnswer




examples - TheAnswer.kt [TheAnswer.app.main]

Project: examples > TheAnswer > app > src > main > java > course > examples > theanswer > TheAnswer.kt

```
1 package course.examples.theanswer
2
3 import ...
4
5 class TheAnswer : AppCompatActivity() {
6
7     companion object {
8         private val answers = arrayOf(42, -10, 0, 100, 1000)
9         private const val answer = 42
10        private const val TAG = "TheAnswer"
11    }
12
13    override fun onCreate(savedInstanceState: Bundle?) { savedInstanceState: null
14
15        // Required call through to Activity.onCreate()
16        // Restore any saved instance state
17        super.onCreate(savedInstanceState) savedInstanceState: null
18
19        // Set up the application's user interface (content view)
20        val binding = AnswerLayoutBinding.inflate(layoutInflater) binding: AnswerLayoutBinding@20247
21        setContentView(binding.root) binding: AnswerLayoutBinding@20247
22
23        val value = findAnswer()
24        val output =
25            if (value == answer) answer.toString() else "We may never know"
26
27        // Get a reference to a TextView in the content view
28        // Set desired text in answerView TextView
29        binding.answerView.text = output
30    }
31
32
33
34
35 }
```

Emulator: Pixel 2 API 31 x Pixel 3 API 31 x



Debug: TheAnswer.app

Debugger: Console

Frames

- main@19,567 in group "main": RUNNING
- onCreate:27, TheAnswer (course.examples.theanswer)
- performCreate:8051, Activity (android.app)
- performCreate:8031, Activity (android.app)
- callActivityOnCreate:1329, Instrumentation (android.app)
- performLaunchActivity:3608, ActivityThread (android.app)
- handleLaunchActivity:3792, ActivityThread (android.app)
- execute:103, LaunchActivityItem (android.app.servertransaction)
- executeCallbacks:135, TransactionExecutor (android.app.servertransaction)
- execute:95, TransactionExecutor (android.app.servertransaction)
- handleMessage:2210, ActivityThread\$H (android.app)
- dispatchMessage:706, Handler (android.os)
- loopOnce:201, Looper (android.os)
- loop:288, Looper (android.os)
- main:7839, ActivityThread (android.app)
- invoke:-1, Method (java.lang.reflect)
- run:548, RuntimeInit\$MethodAndArgsCaller (com.android.internal.os)
- main:1003, ZygoteInit (com.android.internal.os)

Variables

- this = (TheAnswer@20246) course.examples.theanswer.TheAnswer@4d958bc
- savedInstanceState = null
- binding = (AnswerLayoutBinding@20247)

Launch succeeded

# Development Tools

Android Studio provides numerous tools for monitoring application behaviors

# Example Tools

Device File Explorer

Logcat

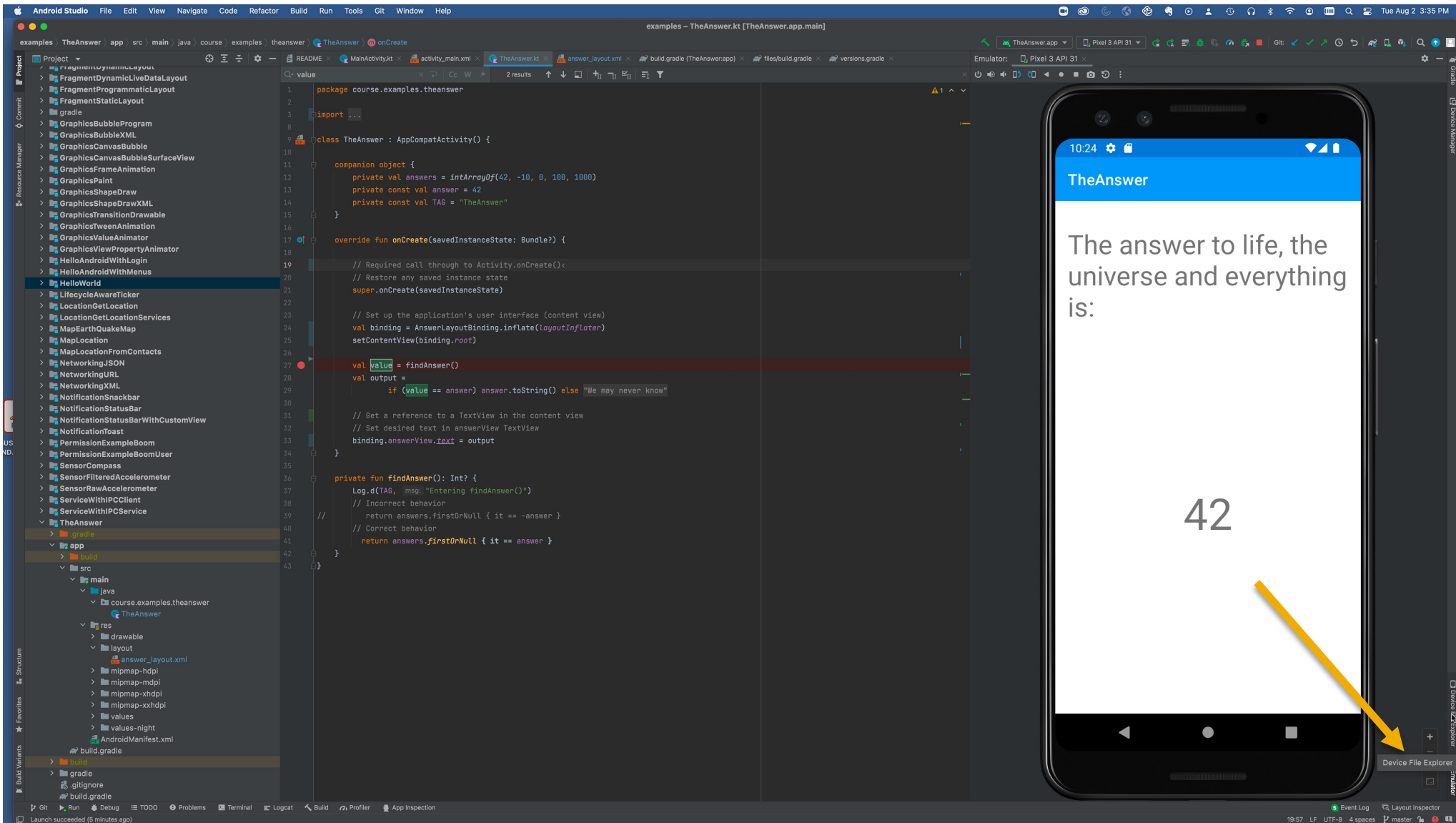
Layout Inspector

CPU Profiler

# Device File Explorer

View, copy, and delete files on your device

Often used to examine and verify file creation and transfer



Android Studio interface showing the code editor for `TheAnswer.kt` and the Device File Explorer.

```
1 package course.examples.theanswer
2
3 import androidx.appcompat.app.AppCompatActivity
4
5 class TheAnswer : AppCompatActivity() {
6
7     companion object {
8         private val answers = intArrayOf(42, -10, 0, 100, 1000)
9         private const val answer = 42
10        private const val TAG = "TheAnswer"
11    }
12
13    override fun onCreate(savedInstanceState: Bundle?) {
14        // Required call through to Activity.onCreate()
15        // Restore any saved instance state
16        super.onCreate(savedInstanceState)
17
18        // Set up the application's user interface (content view)
19        val binding = AnswerLayoutBinding.inflate(layoutInflater)
20        setContentView(binding.root)
21
22        val value = findAnswer()
23        val output =
24            if (value == answer) answer.toString() else "We may never know"
25
26        // Set a reference to a TextView in the content view
27        // Set desired text in answerView TextView
28        binding.answerView.text = output
29    }
30
31    private fun findAnswer(): Int? {
32        Log.d(TAG, msg="Entering findAnswer()")
33        // Incorrect behavior
34        return answers.firstOrNull { it == -answer }
35        // Correct behavior
36        return answers.firstOrNull { it == answer }
37    }
38
39 }
```

Device File Explorer (Emulator Pixel\_3\_API\_31):

File Name	Permissions	Date	Size
com.google.android.gms	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.gms.common	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.gms.common.data	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.gms.location	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.gms.maps	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.gms.maps.model	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.gms.tasks	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.gsf	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.hotspot2.loslogin	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.inputmethod.latin	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.markup	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.modulemetadata	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.networkstack	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.networkstack.permissionconfig	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.networkstack.tethering	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.networkstack.tethering.emulator	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.onetimeinitializer	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.overlay.emulator.config	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.overlay.google.config	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.overlay.permissioncontroller	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.overlay.pixel.config.common	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.packageinstaller	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.partnersetup	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.permissioncontroller	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.printservice.recommendation	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.projection.gearhead	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.providers.media.module	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.sdksetup	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.settings.intelligence	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.setupwizard	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.soundpicker	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.syncadapters.contacts	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.tag	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.tts	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.webview	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.wifi.resources	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.youtube	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.android.telemetry	drwxrwx--x	2022-08-01 10:43	4 KB
com.google.pixel.exo	drwxrwx--x	2022-08-01 10:43	4 KB
course.examples.contentproviders.consumer	drwxrwx--x	2022-08-01 10:43	4 KB
course.examples.contentproviders.provider	drwxrwx--x	2022-08-01 10:43	4 KB
course.examples.helloworld	drwxrwx--x	2022-08-01 10:43	4 KB
course.examples.location.getlocation	drwxrwx--x	2022-08-01 10:43	4 KB
course.examples.location.getlocation.services	drwxrwx--x	2022-08-01 10:43	4 KB
course.examples.theanswer	drwxrwx--x	2022-08-01 10:43	4 KB
org.chromium.webview_shell	drwxrwx--x	2022-08-01 10:43	4 KB
local	drwxrwx--x	2022-08-01 10:43	4 KB
debug_ramdisk	drwxr-xr-x	2008-12-31 19:00	4 KB
dev	drwxr-xr-x	2022-08-01 10:43	2.7 KB
etc	lrw-r--r--	2008-12-31 19:00	11 B
lost-found	drwx-----	2008-12-31 19:00	16 KB
mnt	drwxr-xr-x	2022-08-01 10:43	320 B
odm	drwxr-xr-x	2008-12-31 19:00	4 KB
odm_dtkm	drwxr-xr-x	2008-12-31 19:00	4 KB
oem	drwxr-xr-x	2008-12-31 19:00	4 KB
proc	dr-xr-xr-x	2022-08-01 10:43	0 B
product	drwxr-xr-x	2008-12-31 19:00	4 KB
sdcard	lrw-r--r--	2008-12-31 19:00	21 B
second_stage_resources	drwxr-xr-x	2008-12-31 19:00	4 KB
storage	drwx--x---	2022-08-01 10:43	100 B
sys	dr-xr-xr-x	2022-08-01 10:43	0 B
system	drwxr-xr-x	2008-12-31 19:00	4 KB
system_ext	drwxr-xr-x	2008-12-31 19:00	4 KB
vendor	drwxr-xr-x	2008-12-31 19:00	4 KB
vendor_dtkm	drwxr-xr-x	2008-12-31 19:00	4 KB
bugreports	lrw-r--r--	2008-12-31 19:00	50 B

# Logcat

Write and review log messages

Apps use Log class to write messages to log

Developer can search and filter log messages



The screenshot displays the Android Studio IDE with the following components:

- Project Structure:** A tree view on the left showing the project hierarchy, including folders for resources, assets, and the 'app' module.
- Code Editor:** The main area showing the Kotlin code for 'TheAnswer.kt'. The code includes:

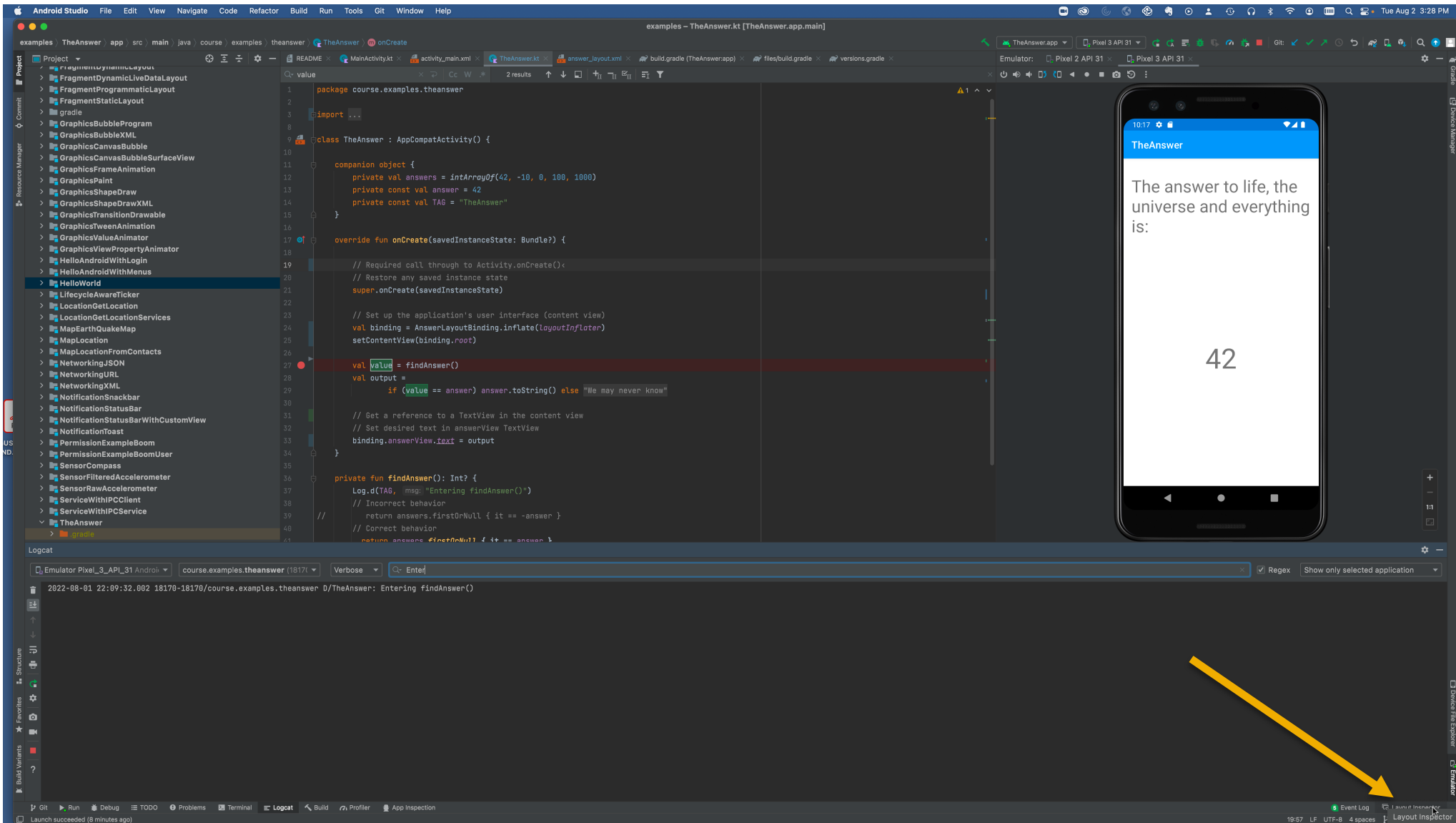
```
1 package course.examples.theanswer
2
3 import ...
4
5 class TheAnswer : AppCompatActivity() {
6
7     companion object {
8         private val answers = intArrayOf(42, -10, 0, 100, 1000)
9         private const val answer = 42
10        private const val TAG = "TheAnswer"
11    }
12
13    override fun onCreate(savedInstanceState: Bundle?) {
14        // Required call through to Activity.onCreate()
15        // Restore any saved instance state
16        super.onCreate(savedInstanceState)
17
18        // Set up the application's user interface (content view)
19        val binding = AnswerLayoutBinding.inflate(layoutInflater)
20        setContentView(binding.root)
21
22        val value = findAnswer()
23        val output =
24            if (value == answer) answer.toString() else "We may never know"
25
26        // Get a reference to a TextView in the content view
27        // Set desired text in answerView TextView
28        binding.answerView.text = output
29    }
30
31    private fun findAnswer(): Int? {
32        Log.d(TAG, msg: "Entering findAnswer()")
33        // Incorrect behavior
34        return answers.firstOrNull { it == -answer }
35        // Correct behavior
36        return answers.firstOrNull { it == answer }
37    }
38}
```
- Emulator:** A virtual Android phone on the right displaying the app's UI. The screen shows the title 'TheAnswer' and the text 'The answer to life, the universe and everything is: 42'.
- Logcat:** A log viewer at the bottom showing system logs and app-specific messages, including 'Entering findAnswer()'.

The screenshot displays an IDE environment with the following components:

- Project Explorer (Left):** Shows a project structure for 'TheAnswer' with various Android components like 'HelloWorld', 'LifecycleAwareTicker', and 'TheAnswer'.
- Code Editor (Center):** Contains Kotlin code for 'TheAnswer.kt'. The code defines a companion object with a list of numbers and a 'findAnswer()' function that returns 42. The 'onCreate()' method sets up the UI to display the result.
- Mobile Emulator (Right):** Shows a virtual Android phone displaying the app's output: 'The answer to life, the universe and everything is: 42'.
- Logcat (Bottom):** Shows a log message: '2022-08-01 22:09:32.882 18170-18170/course.exampLes.theanswer D/TheAnswer: Entering findAnswer()'.

# Layout Inspector

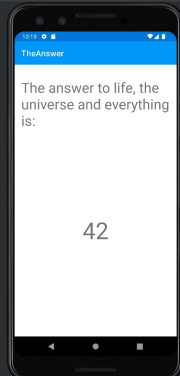
Shows the runtime organization of the user interface



examples - TheAnswer.kt [TheAnswer.app.main]

```
1 package course.examples.theanswer
2
3 import androidx.appcompat.app.AppCompatActivity
4
5 class TheAnswer : AppCompatActivity() {
6
7     companion object {
8         private val answers = intArrayOf(42, -10, 8, 168, 1680)
9         private const val answer = 42
10        private const val TAG = "TheAnswer"
11    }
12
13    override fun onCreate(savedInstanceState: Bundle?) {
14        // Required call through to Activity.onCreate()
15        // Restore any saved instance state
16        super.onCreate(savedInstanceState)
17
18        // Set up the application's user interface (content view)
19        val binding = AnswerLayoutBinding.inflate(layoutInflater)
20        setContentView(binding.root)
21    }
22
23
24
25
26
```

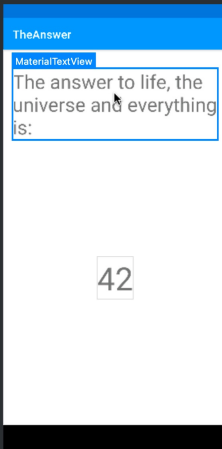
Emulator: Pixel 3 API 31



Layout Inspector

Component Tree

- ConstraintLayout
  - Ab text\_view - "The answer to life, the universe and everything is:"
  - Ab answer\_view - "42"



Attributes

Ab MaterialTextView	
x	16dp
y	112dp
width	360dp
height	125dp
<b>Declared Attributes</b>	
> id	@id/text_view
> text	The answer to life, the universe and everything is:
<b>Layout</b>	
layout_width	wrap_content
layout_height	wrap_content
layout_marginLeft	0dp
layout_marginTop	16dp
layout_marginRight	0dp
layout_marginBottom	0dp
<b>All Attributes</b>	
accessibilityFocused	false
accessibilityHeading	false
accessibilityLiveRegion	none
activated	false
alpha	1.0
autoLink	
autoSizeMaxTextSize	-1
autoSizeMinTextSize	-1
autoSizeStepGranularity	-1
autoSizeTextType	none
baseline	100
> breakStrategy	high_quality
clickable	false
contextClickable	false

**Next**

Application Fundamentals

# Example Applications

HelloWorld

TheAnswer