



# CS501 Mobile Application Development

Fall 2023

Lecture 1

Ron Czik

# What we'll do tonight



Administrivia/course overview



Introduction and ice breakers



Mobile environment



First Android application

# Administrivia



If you are overloaded or have limited time, please drop and retake when you have more availability



Bring your laptops but close them during lecture/discussion

Strongly recommend getting a physical device, they are cheap now  
Run the emulator – more later



Attendance, collaboration, and participation are mandatory

Use Piazza to help each other

- <https://piazza.com/bu/fall2023/cs501>
  - Small amount of extra credit for those who post often and provide good answers
- No makeups



Work in groups on assignments

Specify who you worked with

T E A M W O R K



# This course isn't for everyone

- This is a 500 level, experiential course
  - LOTS of programming
  - We will model industry development practices (Agile/Scrum)
  - Expectations are high and you're required to collaborate
- This course isn't for you if:
  - You don't have time
  - Not able to work in groups
  - Don't have a strong programming background (control structures, looping, arrays, object-oriented techniques)



# Teamwork IS for everyone

- You will all be working together, very closely
- You will be expected to meet and collaborate regularly (several times per week outside of class)
- You are not allowed to coast and receive the same grade as your teammates
  - If you do, you will be graded separately



T E A M W O R K



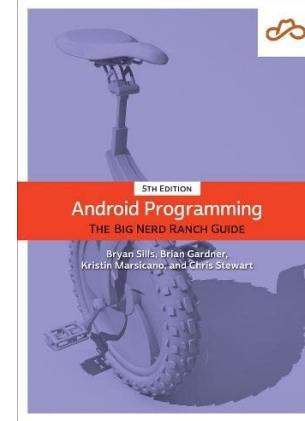
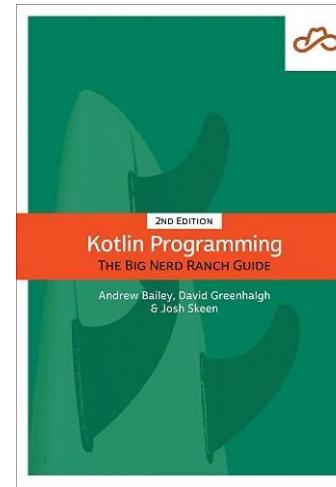
# We will use Kotlin not Java

- Is an open-source, statically-typed programming language that supports both object-oriented and functional programming
- Similar in syntax and concepts to C# and Java
- Managed by the Kotlin Foundation, a group created by JetBrains and Google
- Officially supported by Google for Android development, meaning that Android documentation and tooling is designed with Kotlin in mind
- Certain Android APIs, like Android KTX, are Kotlin-specific, but most are written in Java and can be called from either Java or Kotlin.
- Kotlin's interoperability with Java is core to its growth, meaning that you can call into Java code from Kotlin and vice-versa, leveraging all your existing Java libraries
- We'll review Kotlin next week



# Textbooks and resources

- Android Programming –The Big Nerd Ranch Guide, 5<sup>th</sup> edition – REQUIRED
- Android SDK (<https://developer.android.com/studio>) - REQUIRED
- Kotlin Programming – The Big Nerd Ranch Guide, 2<sup>nd</sup> edition – optional
- <https://kotlinlang.org/>



# Commit!

- Be ready to commit significant effort and time to this course
  - LOTS of new material (this is the first time using Kotlin)
  - LOTS of independent research – there is a lot to Android and we can't cover all of it in 2.75 hours of class time each week
  - The IDE can be clunky – be patient
- Be considerate
  - Do not use electronics while I'm lecturing or while others are presenting – that's just rude!
    - It'll also effect your grade
    - Pay attention, be engaged
- Be accommodating
  - Work closely with your teammates
  - Be flexible, polite, and helpful
- Have high expectations of yourself and your teammates
  - No coasting
  - Everyone contributes – if not, see me
  - I will be checking



# How to succeed?

- Come class prepared every week
  - Read and review the book
  - Be on time
- Be helpful – share ideas
- Listen!
- Meet regularly outside of class
- Be nice
- Work hard





# Introduction and ice breakers

- About me
  - I'm the director of technology digital strategy at the MIT Museum
  - BA in astronomy and MS in computer science, both from BU
  - I've been in the SW industry for 40 years, working at large and small companies
  - I bring the real-world experience into the classroom
  - Education philosophy
    - Collaboration and ongoing assignments – not testing
    - Collaboration on homework assignments isn't cheating – if everyone contributes
  - Love what I'm doing

# Ice breaker

---

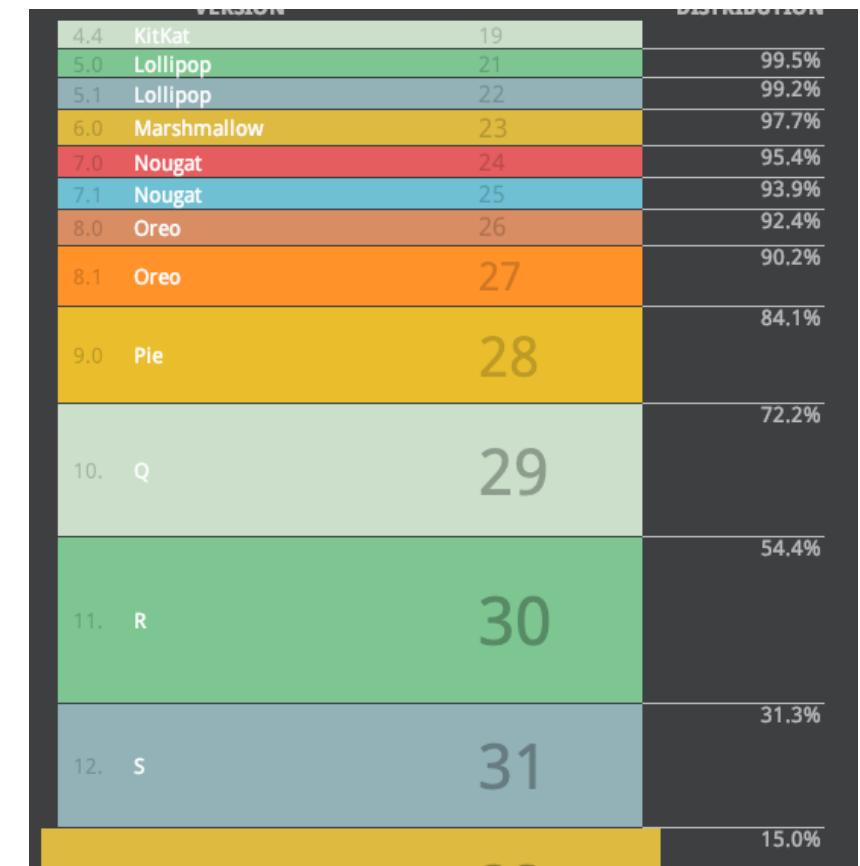
- About you
  - Your name
  - Your hometown and something interesting about it
  - One thing you wish you knew how to do or one thing you recently learned how to do
  - One thing that's interesting about you
  - What you hope to get from this course
  - Any project ideas you have already



# Statistics

---

- Android market is highly fragmented
  - Easiest way to see this is by creating a new project in the IDE
  - After selecting project type, click on “Help me choose”
- Market penetration, Android vs. iOS vs. Win vs. Blackberry
  - <http://www.kantarworldpanel.com/global/smartphone-os-market-share/>



## Course objectives

- Hands on development of Android apps
- Implement small apps at first then transition to semester project
- Utilize the core Android technologies including UI development, action bars, touch, gestures, DB, etc.

## Grading

- |                                     |     |
|-------------------------------------|-----|
| • Assignments and research projects | 30% |
| • Midterm                           | 25% |
| • Final project                     | 35% |
| • Attendance/participation          | 10% |

# Syllabus

# Course approach

Didactic at first

Research

Min-projects/mini-research

Learn how to develop  
Android apps using  
standard industry  
techniques (Agile  
teams)

Android life cycle

Event driving UI

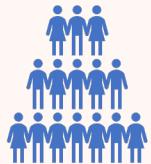
Activities, fragments,  
GPS, Google maps,  
sensors

Learn by doing

# Course expectations



**You will need to come to class prepared – do the research**



**You will work within Agile teams (more later)**

Every team will have a Scrum master



**Everyone participates and contributes value equally**



**Product not project based**

I will serve as the product owner

You will periodically present your accomplishments



# Mobile Environment

---



# What are examples of mobile devices?

- Three attributes: portability, connectivity, computing power
- Cellphone
  - Portability – high
  - connectivity – high
  - computing power – medium
- Wearable (smart watch)
  - Portability – high
  - connectivity – medium
  - computing power – low
- Tablet
  - Portability – medium
  - connectivity – medium/high
  - computing power – high
- Hand-held function specific
  - Portability – high
  - connectivity – high/medium
  - computing power - medium



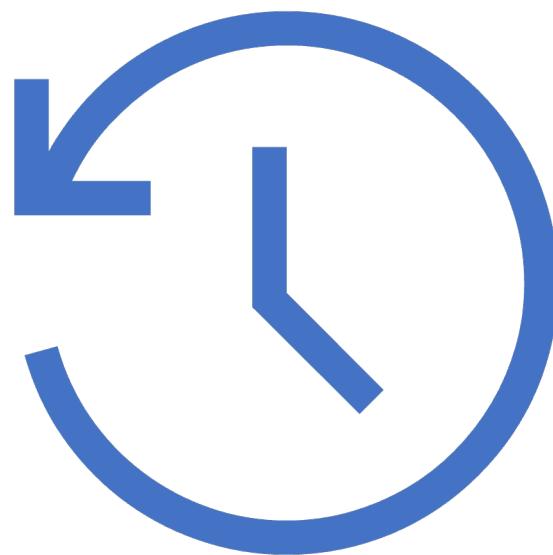
# What is a mobile computer?

- Computational machine
- Portable
- Mobile in its nature of use
- Portable power



# Mobile history

---



What was  
the first  
mobile  
computer?



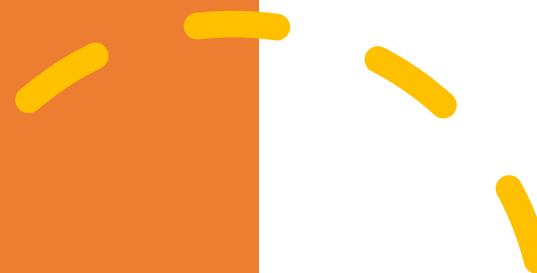
Mobile computing (~500 BCE)

What was  
the first  
mobile  
phone?



Motorola DynaTAC Prototype (1973)

What was  
the first  
commercially  
available  
mobile  
phone?



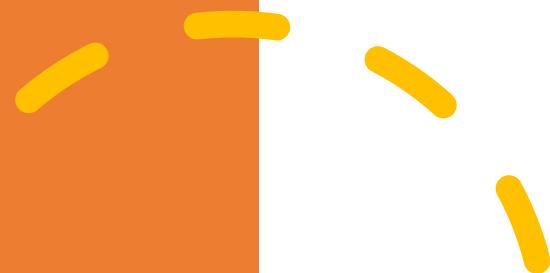
Motorola DynaTAC 8000X (1983)



Nokia Mobira Cityman (1987)

What  
phone did  
Gorbachev  
make  
famous?

What was  
the first  
phone  
'widely  
accessable  
phone?



Motorola Bag Phone (1988)



Motorola MicroTAC (1989)

What was  
the first  
'flip  
phone'?

The first  
GSM (2G)  
phone



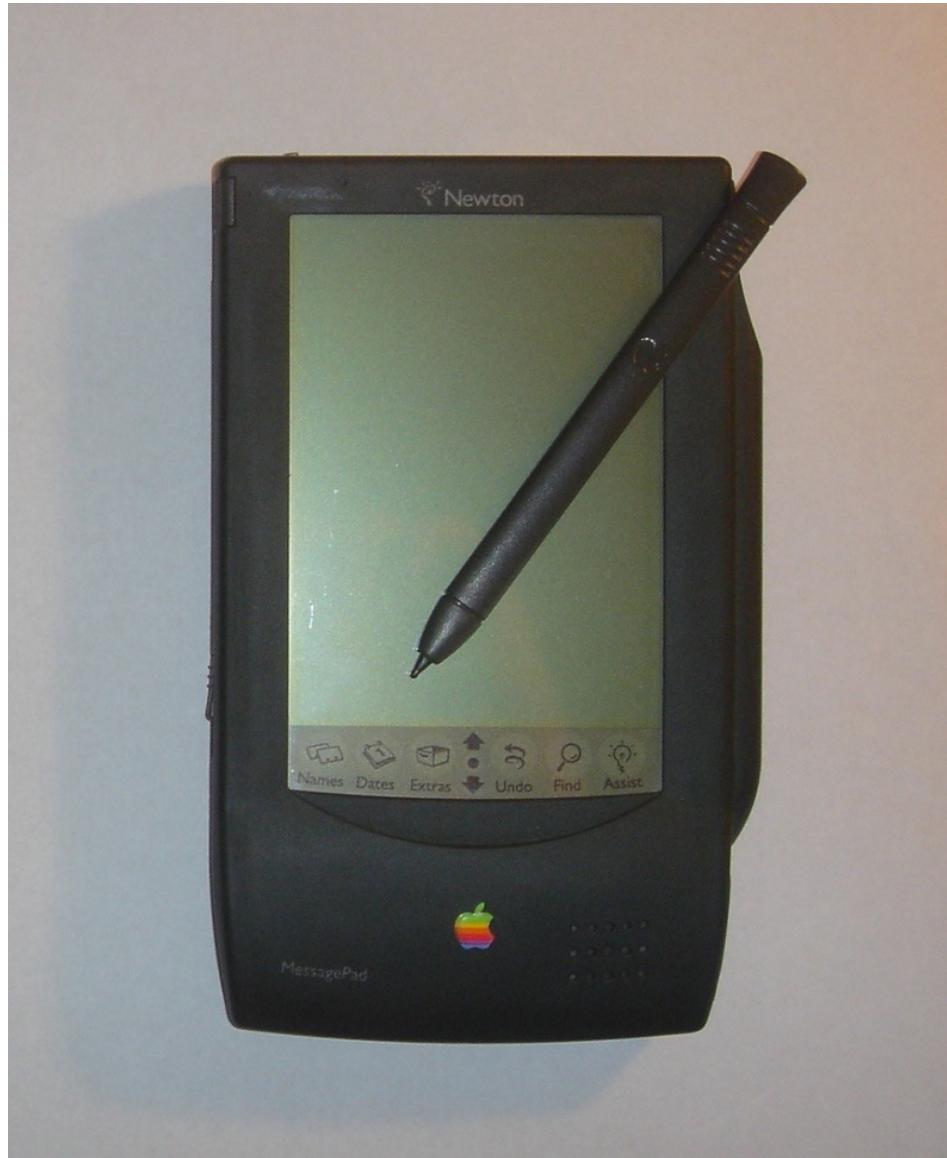
Nokia 101 (1992) – “Candybar phone”

What was  
the first  
touchscreen  
phone?



IBM Simon Personal Communicator (1993)

What  
device  
almost  
killed  
Apple?



Apple Newton MessagePad (1993)

What was the  
first phone  
with a  
keyboard?



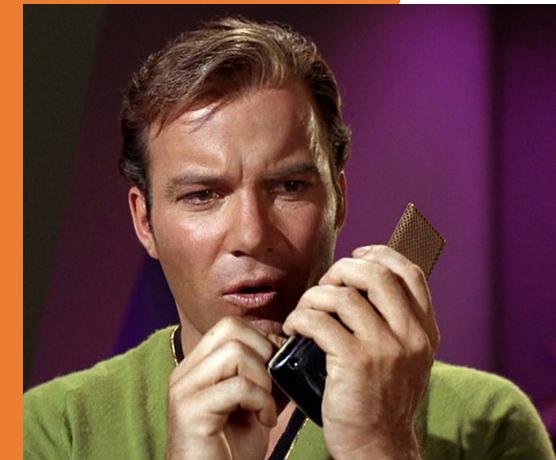
Nokia 9000 Communicator (1996)

What was the  
first successful  
personal digital  
assistant (PDA)?



PalmPilot 5000 (1996)

# What was the 'Star Trek' phone?



Motorola StarTAC (1997)

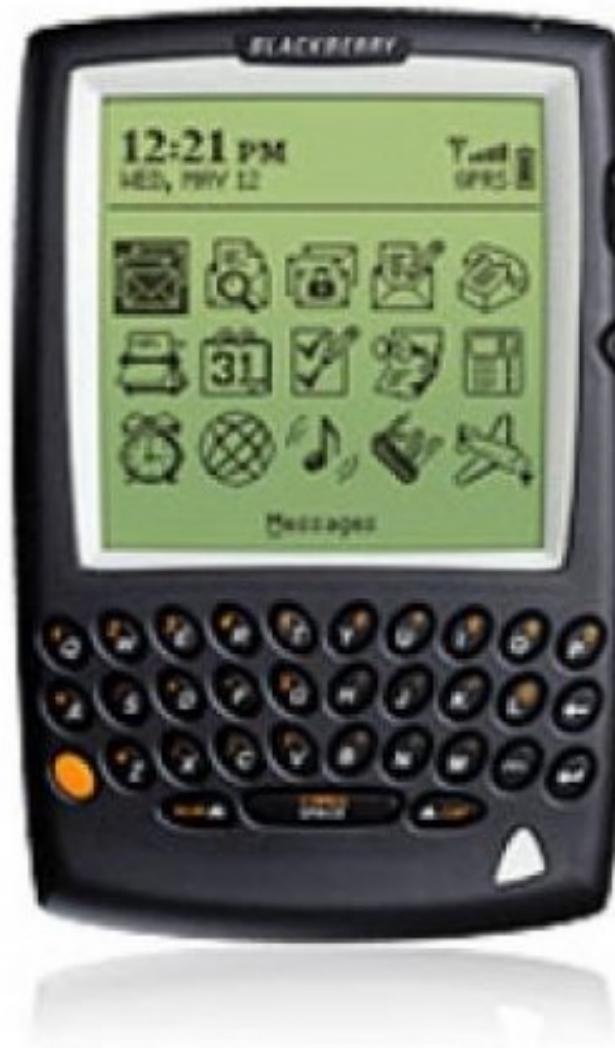


What was  
the first  
'fashion  
phone'?



Nokia 5110 (1998)

What  
phone  
caused  
physical  
injury?



RIM BlackBerry 5810 (2002)

What was  
the device  
that  
combined a  
PDA with a  
phone?



Palm Treo 600 (2003)

What  
phone  
turned  
mobile  
phones into  
a game?



Nokia N-Gage (2003)

# Mobile phone as a weapon?



Motorola Razr V3 (2004)



What was  
the first  
mobile  
phone with  
iTunes?



iTunes Phone - Motorola ROKR E1 (2005)



iPhone



Apple iPhone (2007)



The first  
Android  
phone

HTC Dream (2008)

# Palm OS



Palm Pre (2009)

# The Droid



Motorola Droid (2009)



HTC/Google Nexus One (2010)

The first  
'Google'  
phone

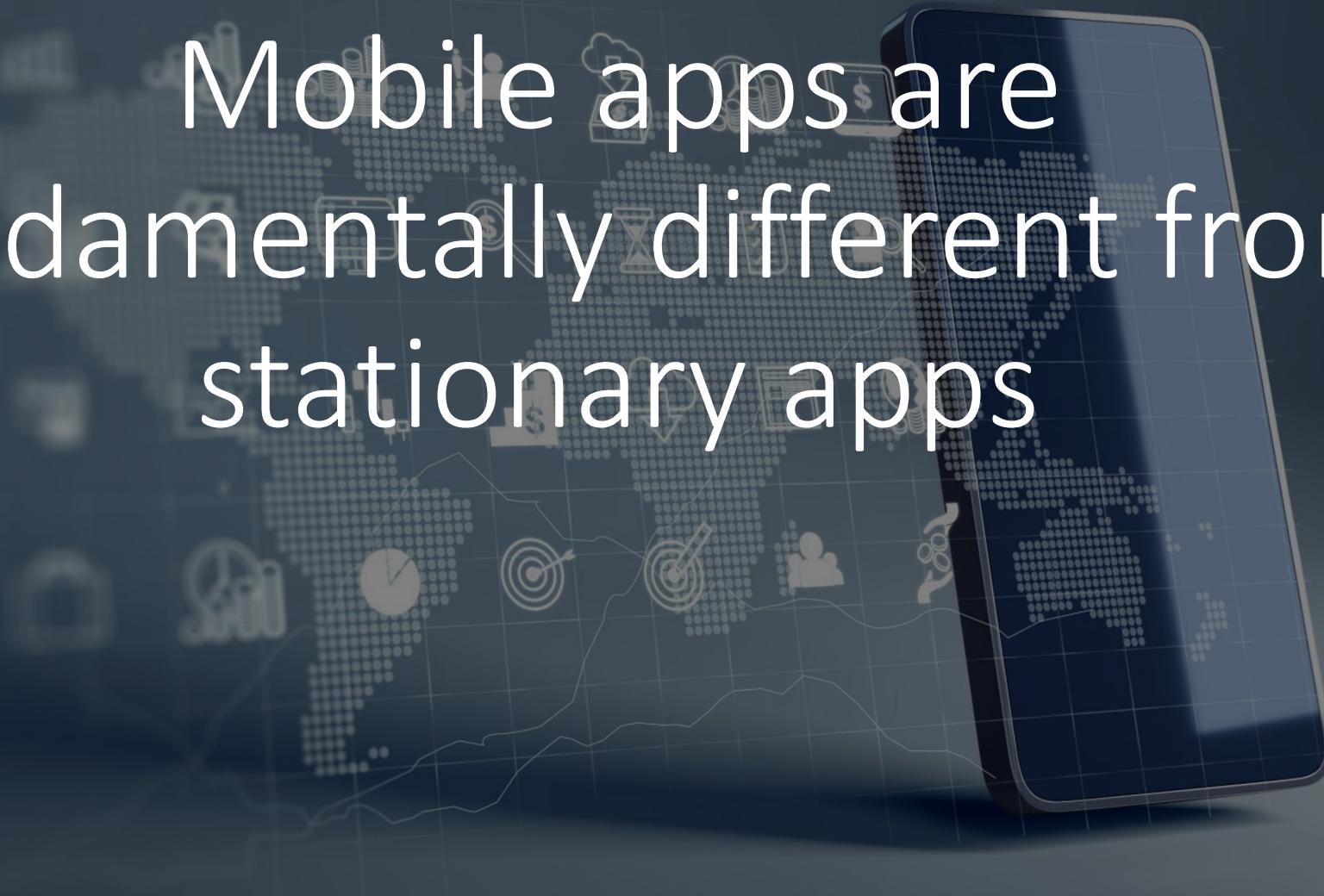
# Android today





Break

Mobile apps are  
fundamentally different from  
stationary apps



# What is mobile computing device?

- A portable computing device that can operate, execute, and provide services and applications
- Is designed to operate without a physical connection (e.g., wirelessly transmit or receive information)
- Is powered-on for extended periods of time with a self-contained power source
- May also include voice communication capabilities, on board sensors that allow the device to capture (e.g., photograph, video, record, or determine location) information, and/or built-in features for synchronizing local data with remote locations

# Examples of mobile computing devices

- Smart phones
- Tablets
- E-readers
- PDAs
- Digital cameras
- Wearable devices
- Personable navigation devices
- Smart cards
- Cars



# Mobile vs stationary applications

- Very different problems
- 4 aspects to consider with mobile computing
  - User
  - Hardware/device
  - Application/purpose
  - Connectivity



➤These 4 considerations determine the ‘mobile condition’

# Some stats

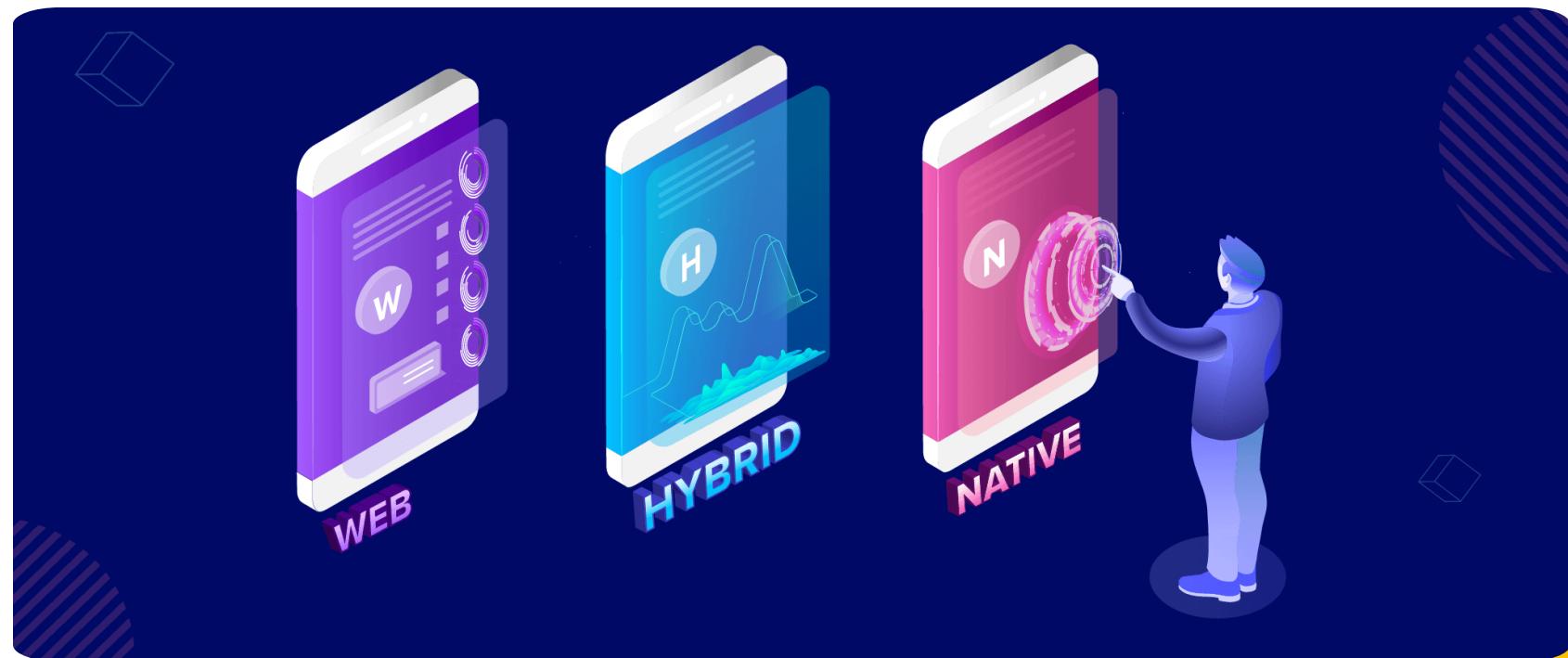
---

- 6.8 billion smartphones (85% of the global population), expected to reach 7.5 billion by 2016
- US adults spend an average of 3:54 on their mobile devices daily (2021)
- Worldwide, 50.9% use mobile phone to purchase online at least once a week (2023)
- 54.8% of all global internet traffic is from mobile devices (2021)
- Smartphones account for 70% of total digital media time (2019)
- Mobile account for 67.9% of total digital ad spent in 2021



# Types of mobile applications

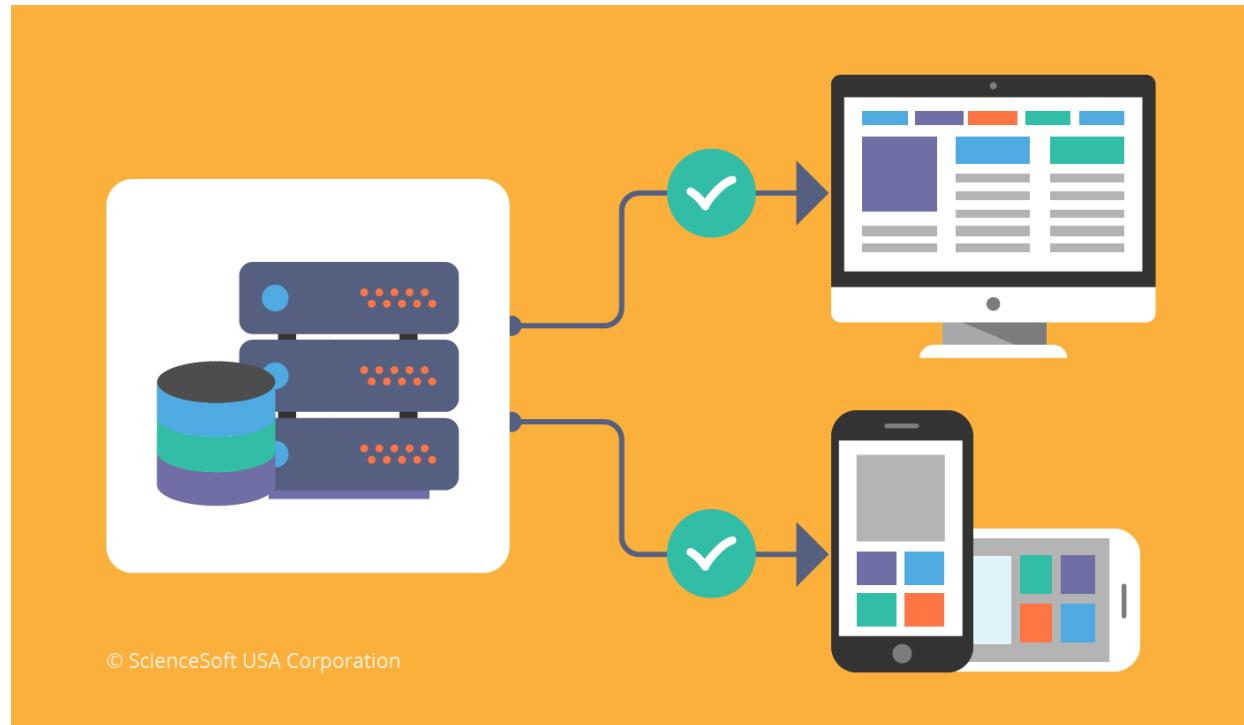
- Web Apps
- Hybrid Apps
- Native Apps



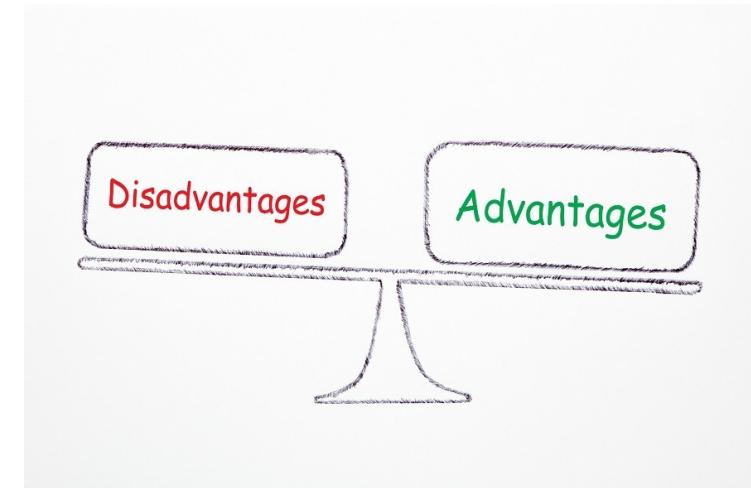
# Web apps

---

- Web apps or mobile web apps work can be accessed from an internet browser window
- They do not require any storage space or installation process to use the app
- Adapt to various screen sizes and devices easily
- Responsiveness and functionality of the web apps could easily be confused with a native app, since both the Native and web apps have almost the same features and responsive nature



# Web app pros and cons



## Advantages

- Reduced business cost
- No installation needed
- Better reach as it can be accessed from anywhere
- Always up-to-date

## Disadvantages

- Web apps fail to work when you are offline
- Limited number of functionalities as compared to Native apps
- It takes a longer time to develop
- Security risk

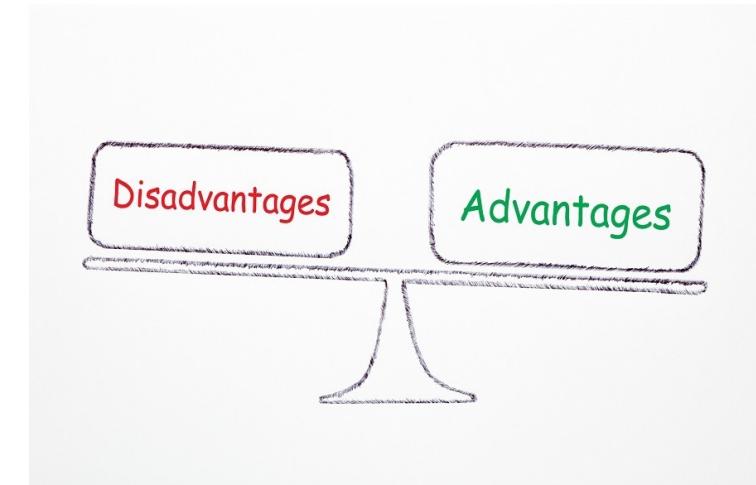
# Native apps

---

- Native Apps are written in a specific programming language to work on a particular Operating system
- Built for specific OS to make the most of the functionalities of the devices that run the particular OS
- Built to make the most of all the features and tools of the phones such as contacts, camera, sensors, etc
- Ensure a high performance and elegant user experience as the developers use the native device UI to build apps



# Native app pros and cons



## Advantages

- Fast performance due to simple code specific to device and OS
- Better use of OS and device specific functionalities
- Interactive UI/UX
- Lesser compatibility issues and faster to configure

## Disadvantages

- Building OS specific apps can be time-consuming
- Need to use OS specific programming languages like Swift and Kotlin
- Longer release cycles to ensure stability
- Requires separate codebase to add new features

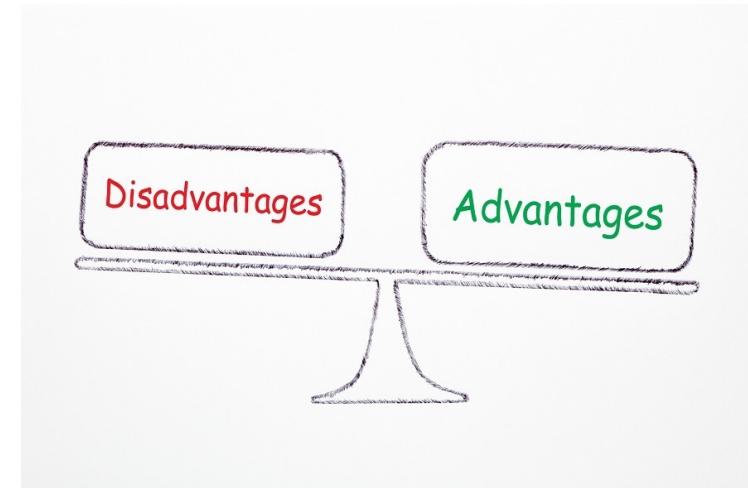
# Hybrid apps

---

- Mix of native and web-based apps. Apps developed using Apache Cordova, Flutter, Xamarin, React Native, Sencha Touch, and other frameworks fall into this category
- These are made to support web and native technologies across multiple platforms
- Moreover, these apps are easier and faster to develop
- It involves use of single codebase which works in multiple mobile operating systems



# Hybrid app pros and cons



## Advantages

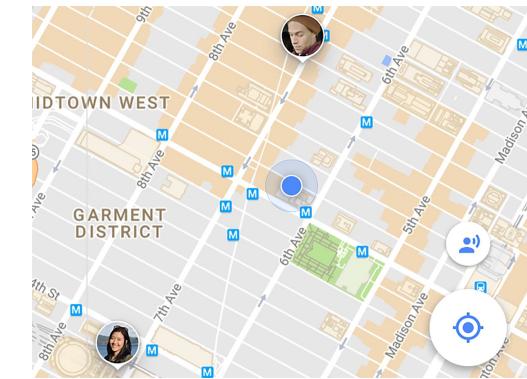
- Easy to build
- Shareable code makes it cheaper than a native app
- Easy to push new features since it uses a single code base
- Can work offline
- Shorter time to market, as the app can be deployed for multiple OSs

## Disadvantages

- Complex apps with many functions will slow down the app
- More expensive than web apps
- Less interactive than native apps
- Apps cannot perform OS specific tasks

# Mobile applications are very personal

- User aware
- Time aware
- Location aware
- Situation aware



- Mobile applications can understand **context**
- Mobile applications know **YOU!**

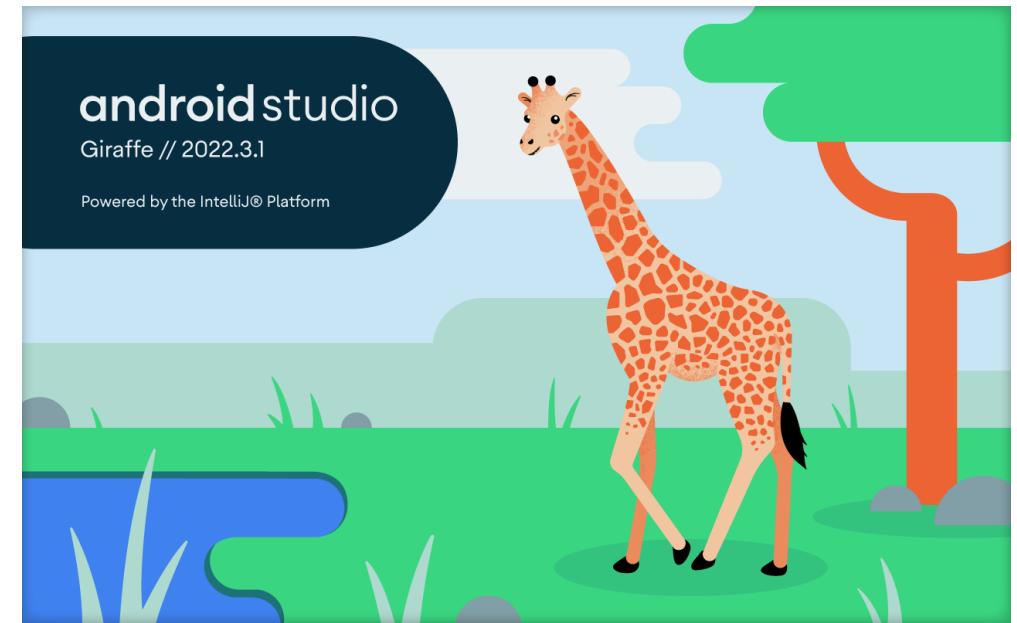
A large, bright green Android robot head is positioned in the upper right corner of the image. It has a simple, rounded design with two grey circular eyes, a small black antenna on top, and a wide, smiling mouth. The body of the robot is also a bright green color.

Starting with Android

# Necessary tools

---

- Android Studio – an integrated development environment used for Android development that is based on IntelliJ IDEA
- An install of Android Studio includes:
  - Android SDK – the latest version of the Android SDK
  - Android SDK tools and platform tools – tools for debugging and testing your apps
  - A system image for the Android emulator – a tool for creating and testing your apps on different virtual devices
- Giraffe – 2022.3.1 – stable release



# Living on the wild side

---

- Beta SDK – Hedgehog
  - Multipreview templates
  - More metrics and crash analytics
  - Device mirroring
  - Debugger enhancements
  - Embedded layout inspector
  - Studio Bot – AI conversational coding support tool – **MUST cite use**

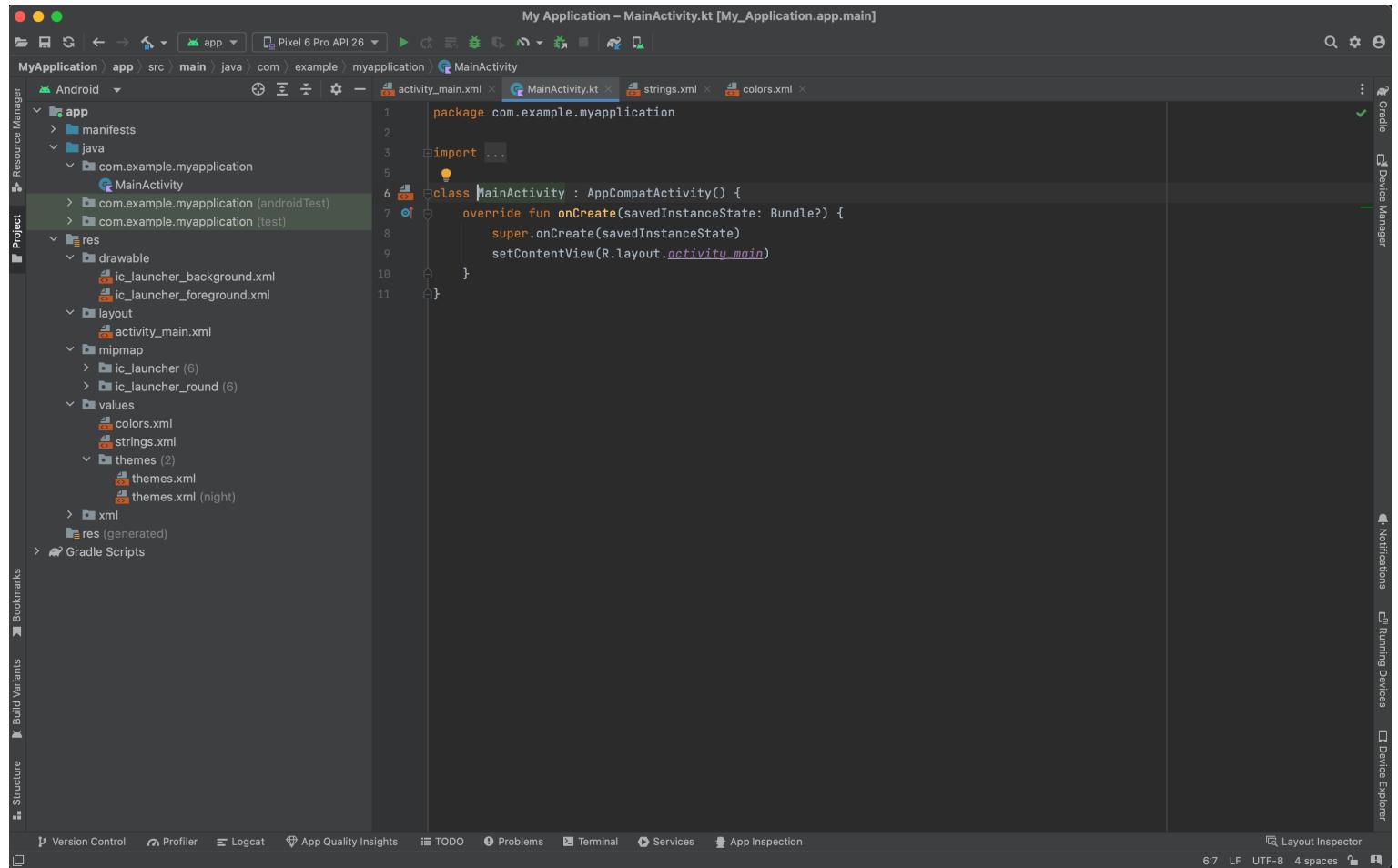


➤ We will be using Giraffe

# Downloading and Installing Android Studio

- Android Studio is available from Android's developer site at [developer.android.com/studio](https://developer.android.com/studio)
  - Includes everything you need to build and run Android applications, including a built-in installation of the Java Development Kit
  - If you want to build and compile Android apps from somewhere outside Android Studio, such as from the command line, you need to have a local installation of the Java Development Kit
    - The latest version of the Android Gradle plugin, which is the tool that builds and compiles Android apps, requires Java 11
    - More recent versions of the Java Development Kit should also work just fine
    - If you are having problems, return to [developer.android.com/studio](https://developer.android.com/studio) for more information

# Explore the SDK – Hello World



The screenshot shows the Android Studio interface with the following details:

- Title Bar:** My Application – MainActivity.kt [My\_Application.app.main]
- Toolbar:** Includes icons for back, forward, search, and other common operations.
- Project Structure:** Shows the project tree under "app".
  - src/main/java/com/example/myapplication/MainActivity:** The current file is MainActivity.kt, which contains the following code:

```
package com.example.myapplication

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```
  - src/res:** Contains layout, drawable, mipmap, and values folders.
  - gradle Scripts:** Shows build.gradle and settings.gradle.
- Bottom Navigation:** Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, Layout Inspector, and Device Explorer.
- Status Bar:** Shows the time (6:7), file type (LF), encoding (UTF-8), and code style (4 spaces).

# Kotlin resources

- There's LOTS more. Here are some resources to review BEFORE next class
  - Kotlin: <https://kotlinlang.org/>
  - Getting started with Kotlin: <https://kotlinlang.org/docs/home.html>
  - Kotlin language specification: <https://kotlinlang.org/spec/introduction.html>
  - Android Kotlin: <https://developer.android.com/kotlin>

# Our first Android app – Hello World

# For next week

- Kotlin
- 30 minute crash course - <https://developer.android.com/kotlin/learn>
- Make sure you signup for Piazza and can access Blackboard
- Assignment 1
- Start thinking about a group you want to create
  - 2-3 people
  - Think about some project proposals