Summer 2, 2019 - CS 4520/CS5520 – Mobile Application Development

Pratheep Kumar Paranthaman, Ph.D.,

About this course

- This course will provide the foundational knowledge on mobile application development.
- you will be learning the interface design, programming, rapid prototyping techniques, project management and **user testing**.

Project-oriented course

Course overview

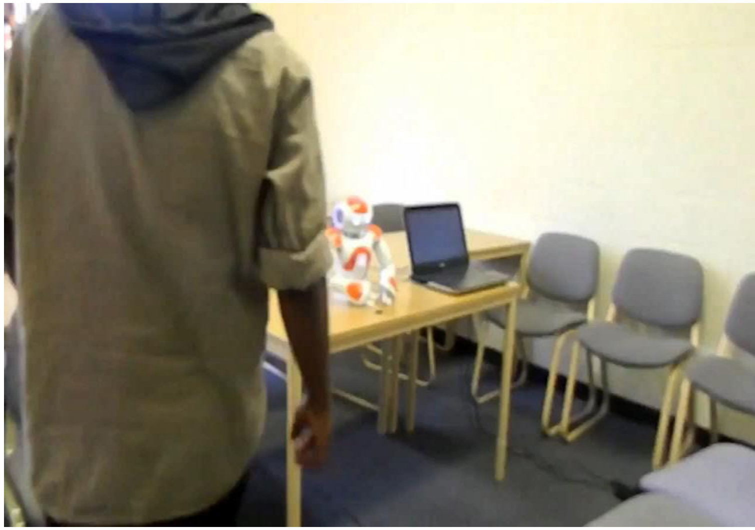
- This course is centered on **Android platform.**
- You will learn and practice the app development lifecycle:

Brainstorming -> prototyping -> developing -> user testing

Who am I?

Background

- ***M.S.*** Artificial Intelligence- Heriot-watt University, U.K.
  - specialization: Human-Robot Interaction
- ***Ph.D.*** University of Genoa, Italy
  - ***Research Topic:*** Serious games for road safety
  - Part of development team in a collaborative EU project on green and safe mobility.



Northeastern!

Research Interests :

- Investigation of player experience in games
- Development of game analytics tools for improving the game development process.
- Use of serious games to improve user behaviour in various contexts.

My availability

- **Office hours:**
  - Wednesdays and Thursdays – between 14:30 - 16:00 @ RY 124
  - **The first office hour will be from next week this Wednesday(7/10) between 2:30 – 4 pm @RY 124**
- Piazza Forum -

  piazza.com/northeastern/summer2019/cs4520cs5520
- email – pratheepk@northeastern.edu
- I can meet during other days as well (depending on my research schedule).

Course Assistants

- Nikitha Preetham
  - Contact: preetham.n@husky.neu.edu
  - Office hours: Monday 4 - 6 PM
  - Location: RY 268

- Piyush Srivastava
  - Contact: shrivastava.pi@husky.neu.edu
  - Office hours: Friday (3 - 5 PM)
  - Location: RY 243

- Ajeya Kempegowda
  - Contact: kempegowda.a@husky.neu.edu
  - Office hours: Tuesday (3 - 5 PM)
  - Location: RY 243

Resources

- **Blackboard class page** – lecture notes, course materials, Piazza signup link, assignments, project and announcements.

- **Books** (below specified books are freely available on Northeastern eLibrary)
  - Burnette, Ed. Hello, Android, 4th Edition (2015). Hello Android
  - DiMarzio, Jerome. Beginning Android Programming with Android Studio, Fourth Edition (2016). Web. Beginning Android Programming
  - Android Studio 2 Essentials – Second Edition (Link)
  - Milette, Greg., Stroud, Adam, and ProQuest. *Professional Android Sensor Programming*(2012). Web. (Link) – **CS5520**

- Lecture notes will be uploaded on Blackboard after each lecture.

Tools that you need for this course

- JDK(Java Development Kit)
- Android Studio – Opensource IDE
  - you can create any app you like. smartphones, tablets, wear, TV apps, and even auto apps

Note: Installation takes longer..

Android Studio Support sessions

- A separate session for installing/ configuring Android Studio for this course will be conducted this week by the TAs:

Location RY 245

Monday(7/1) 3 - 5 PM	**(Nikitha and Ajeya)**
Tuesday(7/2) 3- 5 PM	**(Piyush)**
Wednesday(7/3) 3- 5 PM	**(Nikitha and Ajeya)**

What if I don’t have an Android phone?

- Emulator (good for some parts of the class – assignments don’t a physical device)
- Friends
- I’ll try bringing some Android devices to class(but I’m not 100% sure on this)

- Do I need to Pay for publishing my app on PlayStore?
  - Yes, there is one-time fee of $25 (charged on your Google Developer account)
  - **This course doesn’t expect you to purchase it**, but there are extra credits if you publish your project on the PlayStore.

Resources

- https://freesound.org/
- https://www.iconfinder.com/
- https://pixabay.com/

Make sure to check the license.

What to expect from the course?

Goodbye World

Hello Mobile World

What will you learn?

- Week 1: Introduction to Android and prototyping techniques
- Week 2: Basic GUI and Android Widgets
- Week 3: Android Features, Multimedia and design
- Week 4: data Storage
- Week 5: Web APIs
- Week 6: Localization, sensors and Misc

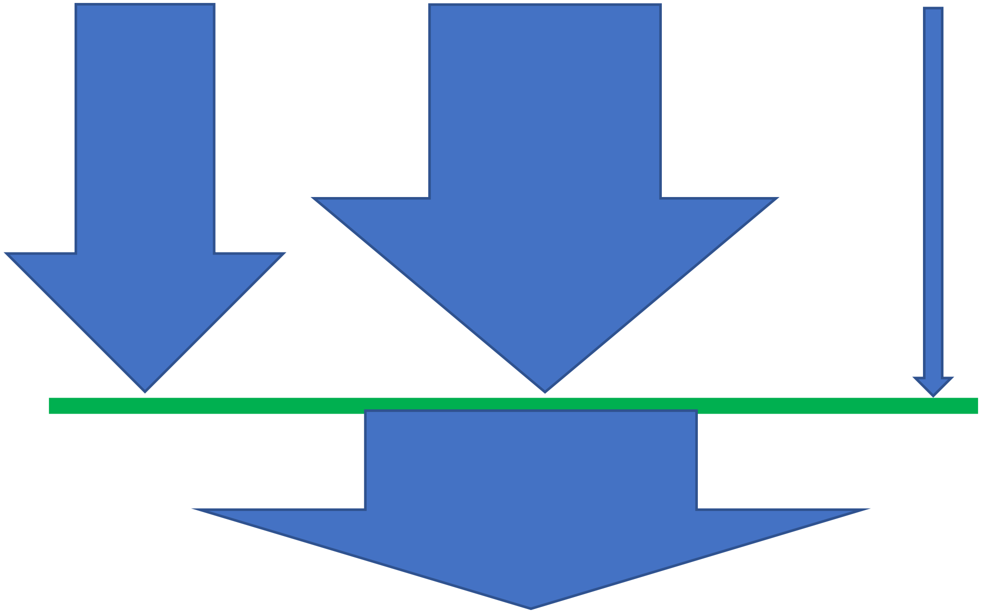
Requirements

- **Weekly**
  - **Research:** Read (and absorb!) – every assignment will have a reading/analysis section
  - **Assignments**: there will be weekly assignment(except for the last 2 weeks of this course).
  - **Project contribution:** Weekly checkpoints for project, so you will be required to collaborate with your teammates and contribute for the project weekly.

My lecture pattern

- Theory
- Demonstration
- In-class exercises

Grading



Assignments

40%

Project

55%

In-class exercises

5%

100%

Assignments

- Assignments grading breakdown

  - **4 assignments in total (40% contribution to the final weighted sum)**

Assignment 1(A01):	7%
Assignment 2(A02):	11%
Assignment 3(A03):	11%
Assignment 4(A04):	11%

Assignments

- Notes for assignment submission
  - **Start working on the assignments as early as possible.**
  - **All assignments are due on the day and time indicated on the schedule. Anything turned in after the deadline will receive a 0.**
  - 1 late day is permitted with 24 hours of extension from the actual deadline.
  - Special circumstances will be handled on a case-by-case basis.
- **Evaluation**
  - Sometimes TAs might call you for in-person evaluation(if there are any issues with your project file: such as running it/dependencies)
- Deadline
  - Assignment deadlines will be on Fridays at 11:59 PM – Check the schedule on the Blackboard for the exact deadline for each assignment

Late policy

1 late day is permitted with 24 hours of extension from the actual deadline (without any deductions).

Applicable only for assignments

Academic Integrity Policy

- All violations of the University academic integrity policy must be reported to the Office of Student Conduct and Conflict Resolution (OSCCR)
- Students who cheat often do so in multiple courses; by reporting all violations to OSCCR, we guarantee that such students are suitably punished

The Short Version:

Share ideas, *not* code or content

In this course, you MUST document where you obtain ideas or code from. You should not be obtaining large chunks of code or program structure from other students or online. **(There is a separate section on this in your project documentation)**

If there is any sign of a breach of the code, you *will* get sent to OSCCR. Period. End of story.

Project

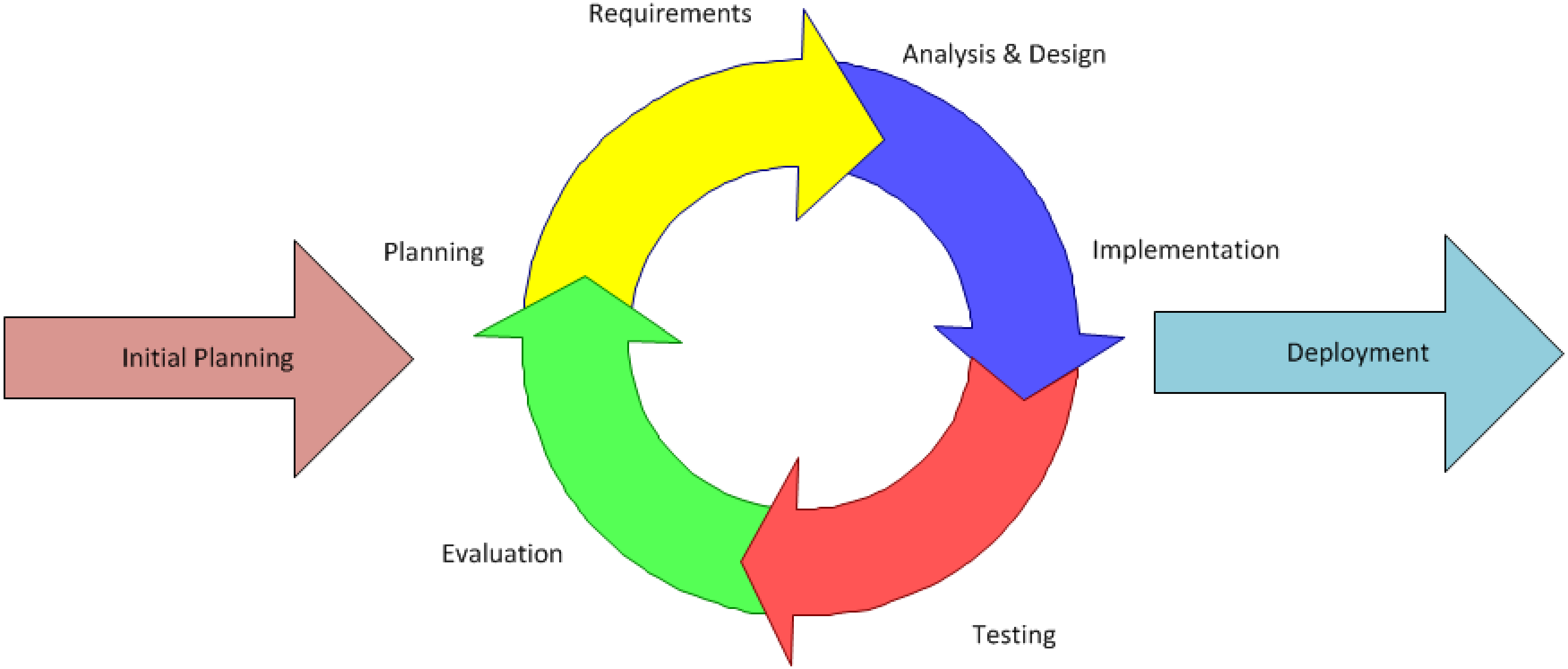
- Work as a part of group and develop a small app in Android Studio.
- Maximum 4 members per group.
- Weekly evaluation(in-class) for project workflow on Thursdays

- **Project grading breakdown – 55% contribution from project**

Project checkpoint1 - proposal and paper prototype	8%
Project Checkpoint 2	9%
Project Checkpoint 3	9%
Project Checkpoint 4	9%
Project Final	10%
Project Presentation	10%

What happens during project checkpoint?

- We will be following an iterative development process to build your project.



[inflectra.com/methodologies/waterfall.aspx]

What happens during project checkpoint?

- Bring your working prototype to class and test it with other team members(**Attendance is mandatory**)
- Either TAs/I will evaluate your project and provide feedback.
- Additionally you get to evaluate your classmates' projects

- How does checkpoints help you with project development?
  - Understand the actual needs early on the development
  - Critical analysis and improvisation on each iteration
  - Reduce the overall burden, as you need to focus only on checkpoints and tune them.
  - Chance to collaborate and critique other apps
  - Better project management.

Project

- Apart from the development, you will also prepare a project documentation

Start from basic proposal -> need classification-> revisions -> final report (you will update this on every milestone)

In-class exercises

- This contributes 5% to the final weighted sum.
- This will be conducted in-class(probably I'll allocate sometime in my lecture for this)
- You can work in pair/alone (it's up to you)
- You can show the output of this either to me or the TAs:
  - In a week I might give a couple of in-class exercises, so you can show it to me after the lecture or at the end of playtesting checkpoints(Thursday).
  - Sometimes I might ask you to replicate the same code that I worked out in-class.
  - I can't say how many in-class exercises I would give in a week, because it depends on my lecture flow and duration of topics.

Where to get help?

- Conceptual doubts and queries
  - Reach me through email and schedule an appointment to discuss
  - Contact the course staffs
- Coding related issues
  - StackOverflow
  - Piazza class forum
  - Just Google!

Versions

- CS 4520
- CS 5520
  - Use sensing capabilities in final project

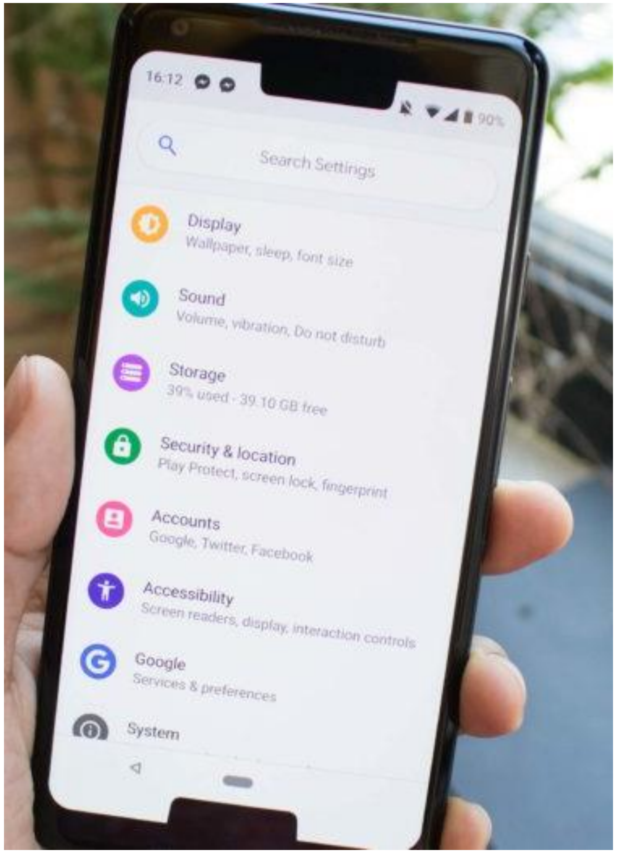
Why? Interference

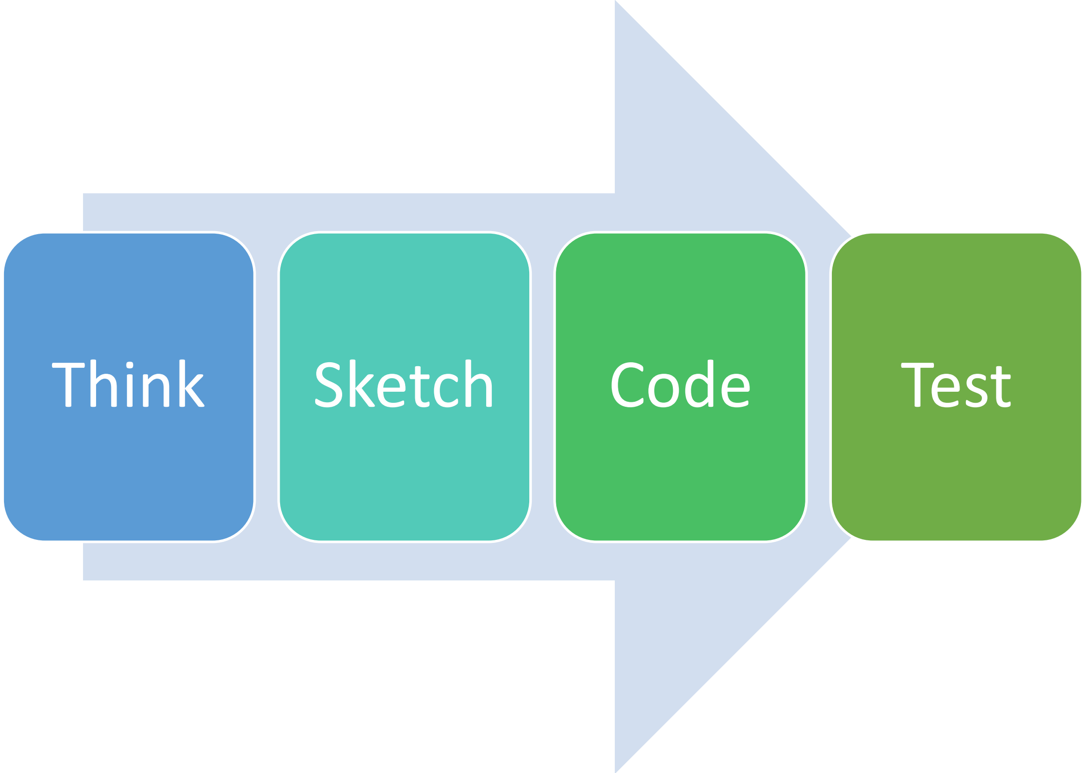
“People who think they are multi-tasking are deluding themselves”

[jp.depositphotos.com/14001432/stock-illustration-cartoon-office-juggler.html]

Let’s start the course now!

Programming an App


16:12
90%
Search Settings
Display
Wallpaper, sleep, font size
Sound
Volume, vibration, Do not disturb
Storage
39% used - 39.10 GB free
Security & location
Play Protect, screen lock, fingerprint
Accounts
Google, Twitter, Facebook
Accessibility
Screen readers, display, interaction controls
Google
Services & preferences
System



Think

Sketch

Code

Test

Mobile Operating System

- Developing application software for low-power devices

Start

getting started
phone
voicemail

1:39 PM 8/2/09
Phone off

text
e-mail
calendar

Contacts Set

[venturebeat.com/2016/01/11/r-i-p-blackberry-os/]

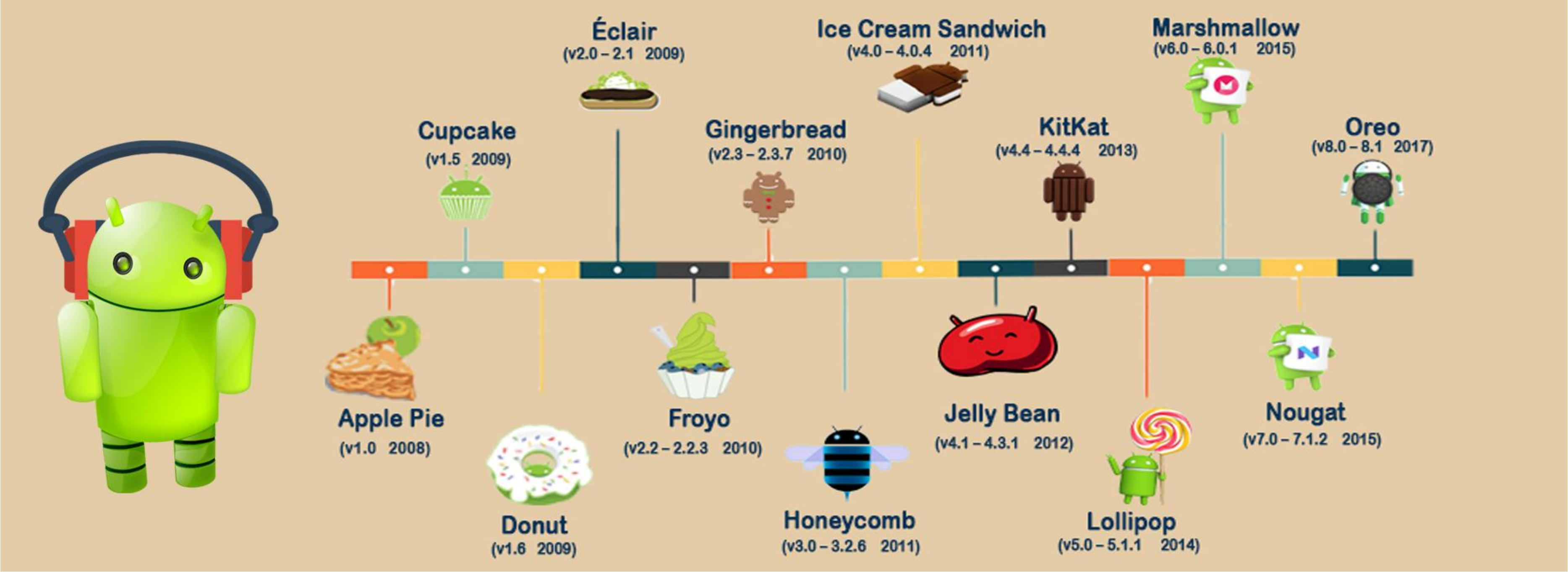
What is Android?

- Mobile operating system based on modified version of Linux.
- Middleware (SDK)
- Originally developed by a start-up company, Android, Inc.
- In 2005 Google purchased Android, Inc. (**OHA - 2007**)
- Google made Android an open source
- Programming based on Java/Kotlin
- Top mobile OS in the world:
  - 2 billion monthly active users
  - 2.6 million apps on Google Play store

[techtimes.com/articles/138921/20160306/verizon-samsung-galaxy-note-5-gets-android-6-0-marshmallow-changelog.htm]

[Wikipedia]

Android Versions



[magivatech.com/blog/info/android-versions]

Android Features

- **Storage**—SQLite, a lightweight relational database, for data storage.
- **Connectivity**—GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), Wi-Fi, LTE, and WiMAX.
- **Messaging**—Both SMS and MMS.
- **Media support** H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
- **Hardware support**—Accelerometer sensor, camera, digital compass, proximity sensor, and GPS.
- **Multi-touch**—Multi-touch screens.
- **Multi-tasking**—Multi-tasking applications.
- **Tethering**—Sharing of Internet connections as a wired/wireless hotspot.

Android Studio

- All you need to build Android Apps!
- Initially the developers used Eclipse IDE and Android Development tools to build apps.
- In May 2013 Google introduced Android Studio, a new development environment based on IntelliJ IDEA.
- Options to build apps for various devices: smartphone, tablet, Wear, TV , and more

[techspot.com/downloads/6831-android-studio.html]

Installing and Configuring Android Studio

- JDK
- Android Studio (Pretty much straight forward process)

Configuring Android Studio:

Maintain a separate folder for this course

- SDK Manager
  - Install SDK platforms(depends on your device / the emulator)
  - System Image(for Emulator)

Emulator (AVD)

- You don’t have to be dependent on physical device.
- Simulates both software and hardware (with some limitations)
- Myriad of devices can be simulated using an Emulator
- Embedded within Android Studio

Can be extremely slow

Android Terminologies

GridLayout

Views

Intents

Emulator

Table View

Manifest file

Textview

Activity

Fragments

Adapters

Gradle

Dalvik Virtual Machine

XML

ListView

Layouts

Content Providers

Let’s create a Demo App

Summary

First and foremost!

Form a team and start preparing for the project checkpoint 1

- Enroll in Piazza
- Assignment 1 will be available on Blackboard(make sure to check it)
- Attend the Android Studio setup session and get help from TAs– in case of any issues with configuring Android Studio in your system

This class is for you if…

- You are an absolute beginner to Android programming(with some programming experience in Java/C#).
- You want to learn to be a mobile app developer.
- You want to challenge yourself to come up with a novel app concept and learn the skills you need to build it so it works robustly.
- You are prepared to make a serious investment in **time and effort** in this class to end up with a great app for your portfolio.
- You want collaborate with fellow developers in class, and learn a few project management skills.

This class is NOT for you if…

- You do not feel comfortable with your Java, C#, or C++ programming skills
- You are not willing to make a major time investment this semester learning Android inside and (even more so) outside class
- You only want to know how to program Android, not **understand development lifecycle of a mobile application**

References

- DiMarzio, Jerome. Beginning Android Programming with Android Studio, Fourth Edition (2016). Web.
- Credits to Prof. Stephen Intille and Prof. Adam C. Champion
- Burnette, Ed. Hello, Android, 4th Edition (2015). Web.