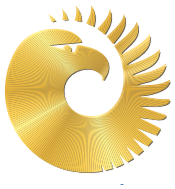

COM-254, Mobile and IoT Application Development

American University of Central Asia
Software Engineering Department

1 Course Description

This course introduces students to the tools, frameworks, and APIs used to build mobile applications for Android and iOS, with a focus on managing networks of physical devices, including vehicles, home appliances, and other embedded systems equipped with sensors, actuators, and other electronics. Students will gain hands-on experience in embedded development using ESP32-based microcontrollers while learning to design and implement interactive user interfaces for multi-touch mobile devices. The curriculum covers object-oriented design principles; Kotlin and Swift; modern development frameworks such as Jetpack Compose and SwiftUI; and device simulators, emulators, and build tools. By the end of the course, students will prototype a real-world mobile application that controls an embedded device and supports and enhances everyday life.

2 Course Information

Course Materials
`https://github.com/auca/com.254`

Course Codes
COM-254

Course ID
4389

Prerequisites
COM-119, Object-oriented Programming

Credits
6

Time and Place

Lecture: Monday 15:35–16:50, Room 410
Lab: Wednesday 15:35–16:50, Room 410

3 Contact Information

Professor

Dmitrii Toksaitov
toksaitov_d@auca.kg

Office

AUCA, Room 315

Office Hours

Office hours are available by appointment, either on-site or remotely, during business hours (Monday–Friday). Please contact your professor to schedule an appointment.

4 Topics

- Weeks 1–2: Overview of Legacy and Contemporary Mobile Platforms; Development Tools (Android Studio, Xcode, SDKs) (6 hours)
- Weeks 3–4: Overview of Swift and Kotlin for Mobile Development (6 hours)
- Week 5: Classic UIKit (iOS) and XML-based (Android) UI Development (3 hours)
- Weeks 6–7: Modern SwiftUI (iOS) and Jetpack Compose (Android) Frameworks for Mobile UI Development (6 hours)
- Week 8: Basics of Mobile Networking (HTTP(S), REST, MQTT) (3 hours)
- Weeks 9–10: Data Storage and Back End (3 hours)
- Week 11: Publishing and Distributing Applications (3 hours)
- Weeks 12–13: Basics of IoT Digital Electronics (6 hours)
- Weeks 14–16: Working with ESP32 Boards (9 hours)

5 Learning Outcomes

By the end of this course, students will:

1. **Acquire basic programming skills to write and test mobile and IoT software**:
    - Develop mobile applications for Android and iOS using modern programming languages like Kotlin and Swift

- Implement interactive user interfaces for multi-touch devices using modern declarative frameworks
- Write embedded software for ESP32-based microcontrollers to control sensors and actuators

2. **Evaluate and use appropriate tools and techniques for mobile and IoT software development**:
    - Utilize industry-standard IDEs like Android Studio and Xcode, including their respective SDKs, simulators, and emulators for development and testing
    - Apply version control practices using Git and GitHub to manage individual and collaborative software projects
    - Work with build tools to prepare and deploy applications for mobile distribution platforms
3. **Design, plan, and critically evaluate mobile and IoT software solutions to problems**:
    - Design and prototype integrated systems that connect a mobile front-end application with a back-end embedded IoT device
    - Apply object-oriented design principles and user-centered design concepts to create intuitive and effective mobile user interfaces
    - Plan the communication architecture between mobile and IoT devices using networking protocols like HTTP/REST or MQTT
4. **Evaluate mobile and IoT systems in terms of quality attributes and trade-offs**:
    - Assess an integrated mobile-IoT system based on criteria such as usefulness, complexity, and level of polish
    - Evaluate the trade-offs between different UI development frameworks in terms of performance and developer productivity
    - Consider the constraints of embedded systems, such as power consumption and processing limitations, when designing IoT components
5. **Reflect and reason about information handling problems**:
    - Analyze and implement strategies for local data storage and persistence on mobile devices
    - Reason about the flow of data between a mobile client, a back-end service, and an IoT device
    - Critically examine methods for collecting, transmitting, and presenting data from sensors in an IoT network
6. **Understand and apply ethical principles and professional standards**:
    - Recognize the ethical implications of mobile and IoT applications, particularly regarding data privacy, security, and user consent
    - Adhere to professional standards for code quality, documentation, and collaborative development throughout the project lifecycle

- Reflect on the societal impact of pervasive mobile and IoT technologies in enhancing everyday life

6 Assignments and Exams

6.1 Moodle/e-Course Checkpoints

Students must maintain instructor-provided private GitHub repositories for their assignments. They must periodically commit and push the required number of lab or project solutions, as directed by the course staff. Instructors or teaching assistants will review the work either during lab sessions (on-site) or after the submission deadline (off-site) and award points based on completed assignments.

6.2 Labs and Projects

Throughout the course, students will complete several laboratory assignments in which they create small mobile apps or embedded programs to explore fundamental concepts in mobile and IoT software development.

Students will also complete a course project. For this project, they will build an integrated system that combines a mobile device and an embedded IoT board working together to serve a useful purpose. The project is open-ended; students may choose the specifics. Evaluation will be based on usefulness, complexity, and a level of polish sufficient for release on mobile app distribution platforms.

7 Course Materials, Recordings and Screencasts

All course materials are available on GitHub at `https://github.com/auca/com.254`. Using GitHub will help students become familiar with the Git version control system and the widely used GitHub platform.

We aim to record each class and upload it to YouTube for accessibility; however, recordings are not guaranteed. Recordings are produced on a best-effort basis as time permits. If you need immediate access, consider recording class sessions on your own computer. Links to YouTube recordings can be found in the course repository at `https://github.com/auca/com.254`.

While recordings provide flexibility, they are not a substitute for attending classes. Active participation is crucial for success in this course. Accumulating three or more unexcused absences may lead to an X grade. If overall class attendance is poor, the instructor reserves the right to discontinue class recordings.

Access lecture screencasts remotely via Zoom at either `http://com-254-zoom.auca.space`. When joining a Zoom session, students must identify themselves using their properly capitalized first and last names in the Latin alphabet. Your lab instructor may also use Zoom or other tools for remote work and set additional etiquette rules. Consult your instructor for more information. Install and configure the required remote tools on your computer during the first week of the semester so you can properly share your camera, microphone, and screen.

Remember that to attend lecture or lab classes remotely, you must obtain written permission from the instructor in advance. If such permission is not granted before

the class begins, you will be marked absent (unexcused) in the attendance system (even if you join via the online platform). Please send a brief email to the instructor explaining why you will attend remotely (e.g., illness, personal reasons) and, when possible, provide appropriate documentation to support your request to participate online, either immediately or later in the same email thread. If appropriate documentation is not provided within a reasonable timeframe in that thread, you will be marked absent without excuse and may be considered for a grade of X if you have three or more unexcused absences. AUAF students who are outside the country and can only participate online do not need to request this permission.

8 Software

Students are recommended to install the following software on their machines.

For those who would prefer to do Android labs and projects:

- Android Studio: `https://developer.android.com/sdk/index.html`
- Android USB driver for ADB (on Windows): `https://developer.android.com/tools/extras/oem-usb.html`

For those who would prefer to do iOS labs and projects:

- Xcode: `https://developer.apple.com/xcode/resources`

For IoT development, everyone will need the following:

- PlatformIO: `https://platformio.org/platformio-ide`

All students should also install:

- Git: `https://git-scm.com/downloads`

9 Hardware

In this course, we require you to have a powerful machine with more memory (rather than less), high-speed disk I/O, a CPU that supports modern virtualization technologies, and a GPU capable of running modern graphics APIs. In addition, having a mobile device would be helpful to debug and test things, not just in an emulator or simulator. Unfortunately, we do not provide lab machines or mobile devices for this course, as it is not a required course. If this is a dealbreaker, please consider changing the course during the add/drop period. As for IoT, we can provide a board for each student and some basic electronics and sensors to prototype an embedded system. You do not need to own or purchase one for the course.

10 Reading

1. The Official Android Developer Guides: `https://developer.android.com/guide`
2. The Official iOS Developer Guides: `https://developer.apple.com`

10.1 Supplemental Reading

1. Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (AUCA Library Call Number: QA 76.64 D47 1995, ISBN: 978-0201633610)

2. Refactoring: Improving the Design of Existing Code by Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts (AUCA Library Call Number: QA76.76.R42 F695 1999, ISBN: 978-0201485677)

11 Grading

The preliminary distribution of points is outlined below. Please note that the distribution may change throughout the course if tasks are canceled, merged, or made optional (for bonus points). This usually happens for reasons such as software issues, online service outages, or classes canceled due to events outside our control.

Remember that some LMS platforms, such as Moodle, may not calculate final scores and grades correctly until all tasks have been published in the system and properly weighted. Do NOT assume your grade until all tasks are in the system and your instructor has notified you to review your scores and grades.

11.1 Moodle/e-Course Checkpoints

Your instructor will periodically announce reviews of your work. For such checks, you can be awarded up to the specified number of points.

- Lab #1 (5%)
- Lab #2 (5%)
- Lab #3 (5%)
- Lab #4 (5%)
- Course Project (20%)

11.2 Exams

In the middle and at the end of the course, you will need to pass the Midterm and Final examinations. These exams are significant and will greatly affect your grade.

- Midterm Exam (25%)
- Final Exam (35%)

11.3 Totals

100% is formed from the Moodle/e-Course Checkpoints (40%) and the two exams (60%).

11.4 Scale

- [94%–100]%: A
- [90%–94)%: A-
- [87%–90)%: B+
- [83%–87)%: B
- [80%–83)%: B-
- [77%–80)%: C+
- [73%–77)%: C
- [70%–73)%: C-
- [67%–70)%: D+
- [63%–67)%: D
- [60%–63)%: D-
- Less than 60%: F

Please note that requests for a higher grade due to points being marginally close will be ignored. For instance, 93.99 is an A-, NOT an A. Similarly, requests for extra assignments to boost points will also be disregarded.

12 Rules

First and foremost, in addition to all the rules listed in the syllabus document, students are required to follow the Code of Conduct of the American University of Central Asia.

12.1 Participation

Active participation during class may be rewarded with extra points at the instructor's discretion. Poor student performance during class can result in points being deducted from the final grade.

Instructors may conduct random pop-up checks during classes without prior notice. Students MUST be prepared for every class to avoid losing points. Students who are absent from classes with graded work without a valid reason will also lose points, unless their absence is due to force majeure circumstances. Instructors must be notified in advance of the reason for a student's absence in order to avoid a loss of points.

12.2 Questions

A question raised by one student is often relevant to others as well. Therefore, students are encouraged to use the online discussion board of the LMS (Learning Management System) they are enrolled in (e.g., AUCA e-Course System) to post questions publicly so that all students may benefit from the discussion.

Students must not post complete source code for any task on the LMS discussion board. Any such public post will result in a grade of zero for that assignment. In

addition, students should refrain from submitting vague or overly general questions such as "Why doesn't my code work?"Instead, they are expected to carefully analyze and debug their code before seeking assistance.

12.3 Late Policy

Late submissions and late exams are not permitted. Exceptions may be granted at the professor's discretion only in cases of force majeure. If you become ill, experience serious personal difficulties, or encounter technical problems with your computer or Internet, you MUST notify the instructors at least 24 hours in advance. Failure to do so will result in no extension being granted. Last-day requests will be considered procrastination. Additionally, no student will be granted more than one emergency extension during the course.

Beginning six hours before the deadline for any assignment, instructors will enter "silent mode."During this period, no questions related to the assignment will be answered, and no requests for office hours will be considered. At all other times prior to the deadline, instructors will make every effort to respond to questions and provide assistance, either via Zoom or in person during office hours.

12.4 Exam and Task Submission Ceremonies

Students MUST adhere to all exam and task submission procedures. This means they must strictly follow the rules specified by the instructors, whether provided in written or verbal form. Failure to comply will result in lost points. In your professional career, you will frequently work with formal documents (e.g., contracts, timesheets, reports), so it is important to develop accuracy and attention to detail early. Points will be deducted—or, in some cases, submissions may be rejected entirely—if the specified procedures are not followed. Any violation of deadlines or strict exam timing rules will result in a grade of zero.

12.5 Administrative Drop

Instructors reserve the right to drop a student from the course for non-attendance. If a student misses three or more classes without an acceptable excuse, the faculty may assign an X grade and remove the student from the course.

12.6 Incomplete Grade

Consistent with the late exam policy, a grade of I (Incomplete) will be granted only under highly exceptional circumstances. Students must initiate the request for an I grade well in advance of the final week of the semester. Requests made during the last week before final exams will not be considered.

12.7 Academic Honesty

Plagiarism is the act of copying or appropriating someone else's words or ideas and presenting them as one's own. In the context of this course, plagiarism may occur in many forms, including but not limited to program code, comments, software

documentation, design specifications, requirement documents, project reports, and technical analyses.

More specifically, examples of plagiarism in a Software Engineering course include:

- Submitting code written by others or generated by AI as your own
- Modifying existing code or designs (e.g., changing variable names or restructuring code) and claiming originality
- Incorporating code snippets, sentences, design patterns, or intellectual content from any source without citation
- Using algorithms, patterns, or architectural designs without proper acknowledgment
- Using graphics, datasets, audio, video, or other materials from external works without attribution
- Purchasing or otherwise acquiring code, software, or project content and presenting it as original work

Plagiarism is unethical and undermines the educational process. The consequences are as follows:

- First instance: A grade of zero for the plagiarized work, and a report filed with the Registrar's Office
- Second instance: A failing grade (F) for the entire course

Both parties involved, the student who plagiarizes and the student whose work was used without authorization, will face equal consequences. It is the responsibility of every student to safeguard their assignments, code, and related materials to prevent unauthorized use.

The use of artificial intelligence, including generative AI tools, to complete certain assignments, projects, or exams (whether off-site or on-site) may be strictly prohibited. If AI involvement is suspected in submitted work, the student may be required to replicate the same or a similar task and respond to related questions in a supervised setting. Failure to do so satisfactorily will be regarded as evidence of AI-generated work, and the penalties described above will apply. Please note that for some tasks, the instructor may permit the use of AI technologies, but explicit written permission must be obtained in the assignment requirements before using them.

Finally, this course adheres to all global university policies on academic honesty. If university-wide regulations are stricter than those outlined here, the stricter rules will take precedence.

13 Access and Support Services

This course is committed to fostering an inclusive learning environment that supports the diverse needs of all students. If you have a disability or require specific accommodations to participate fully, please contact the instructors as early as possible. We will work with you to ensure that appropriate adjustments are made to support your learning experience.

For guidance on time management, presentation skills, writing, or study strategies beyond the scope of Software Engineering, we encourage you to connect with the Advising Office. The Advising Office offers practical workshops and peer advising to help you navigate academic challenges.

If you prefer working with peers, you may schedule an appointment with a student tutor through the Writing and Academic Resource Center (WARC). Tutoring is available both in person and online. Additional resources can be found on the WARC webpage.

If you are feeling stressed, overwhelmed, or struggling with emotions, relationships, or other mental health concerns, we encourage you to seek support from the AUCA Counseling Service. The Counseling Service provides confidential and free professional assistance to students.