COM 254, Mobile and IoT Application Development

American University of Central Asia
Software Engineering Department

1 Course Information

Course Code
COM 254

Course ID
4389

Prerequisite
COM-119, Object Oriented Programming

Credits
6

Professors, TAs, Time, Place
Lecture (Dmitrii Toksaitov): Monday 12:45–14:00, 432
Wednesday (Dmitrii Toksaitov): Wednesday 12:45–14:00, 432

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

Class Discussions
`https://piazza.com/auca.kg/spring2020/com254`

2 Contact Information

Instructor
Toksaitov Dmitrii Alexandrovich
toksaitov_d@auca.kg

Office
AUCA, room 315

Office Hours
By appointment throughout the work week
Remotely through Skype on Saturday and Sunday from 18:00 to 22:00

3 Course Overview

This course introduces students to development tools and APIs to build applications for the Google Android operating system to manage networks of physical devices, vehicles, home appliances, and other items embedded with electronics, sensors, and actuators. Students will get introduced to embedded development on the Arduino platform with the help of ESP8266, ESP32, and several other WiFi and Bluetooth chips with programmable microcontrollers. Students will also learn how to build unique interactive user interfaces for multi-touch mobile devices on the Android platform to manage embedded devices around us. The mobile development part covers object-oriented design, the Java programming language for the Android Runtime, development frameworks, device emulators, and application build tools. Other topics include multi-threading, power and performance considerations, the accelerated 2-D and 3-D graphics APIs. By bringing two platforms together, students will prototype appliances that can be controlled through mobile phones to help people with their daily life. This is an elective course from the base professional cycle of AUCA courses that is designed for Software Engineering majors.

As a result, students should be able to research and analyze the functioning of the information technology systems, improve their skills using programming languages for software design, development, and maintenance in accord to the goals of the AUCA Software Engineering department and the 510300 IT competency standard (including competency elements OK 17, 17, 115).

4 Topics Covered

- Week 1: Development Tools (Android Studio, SDK, Device Emulators) (3 hours)
- Week 2: App. Fundamentals (Activities, Fragments, Services) (3 hours)
- Week 3: User Interface Elements (3 hours)
- Week 4: Graphics and Animation (3 hours)
- Week 5: Data Storage (3 hours)
- Week 6: Connectivity (3 hours)
- Week 7: Media and Camera (3 hours)
- Week 8: Working with Device Sensors (3 hours)
- Week 9: Publishing and distributing applications (3 hours)
- Week 10–11: Basics of Digital Electronics (6 hours)
- Week 12: The Arduino IDE (3 hours)
- Week 13–14: Working with the ESP8266, ESP-32 boards (6 hours)

5 Labs

Students will have to finish 10 lab tasks. In every task students will build a simple android/embedded application.

6 Course Projects

Each student will have to develop an app for the Android platform and a connected device. The challenge of the project is to maintain a certain level of quality for the application to be able to publish it to end users on Google Play Store at the end of the course.

7 Course Materials, Recordings and Screencasts

Students will find all the course materials on GitHub. We hope by working with GitHub students will become familiar with the Git version control system and the popular (among developers) GitHub service. Though version control is not the focus of the course, some course tasks may have to be submitted through it on the GitHub Classroom service.

Every class is screencasted online and recorded to YouTube for students convenience. An ability to watch a class remotely MUST NOT be a reason to not attend the class. Active class participation is necessary to succeed on this course.

8 Reading

1. The Big Nerd Ranch Guide by Bill Phillips, Brian Hardy (ISBN: 978-0321804334)

2. Introduction to Java Programming, Comprehensive, 11th Edition by Y. Daniel Liang (ISBN: 978-0132130806)

8.1 Supplemental Reading

1. Introduction to Android Application Development: Android Essentials, 5th Edition by Joseph Annuzzi Jr., Lauren Darcey, Shane Conder (ISBN: 978-0134389455)

2. 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)

3. 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)

9 Grading

- Practice tasks (60%)
- Course project (40%)
- Piazza Participation (5%)

- 90%–100%: A
- 80%–89%: A-
- 70%–79%: B+
- 65%–69%: B
- 60%–64%: B-
- 56%–59%: C+
- 53%–55%: C
- 50%–52%: C-
- 46%–49%: D+
- 43%–45%: D
- 40%–42%: D-
- Less than 39%: F

10 Rules

Students are required to follow the rules of conduct of the Software Engineering Department and American University of Central Asia.

10.1 Participation

Active work during the class may be awarded with up to 5 extra points at the instructors discretion.

Poor student performance during a class can lead to up to 5 points being deducted from the final grade.

Instructors may conduct pop-checks during classes at random without prior notice. Students MUST be ready for every class in order not to loose points.

10.2 Attendance

Missing more than three classes without a reason will result in 10 points being deducted from the student for every day. If a student has health/family/personal emergency, he MUST notify the instructor in advance (e.g., through e-mail). The student MUST also provide a valid proof afterwards. Without a prior notice and a valid proof the miss will still be counted.

10.3 Questions

We believe that a question from one student is most likely a question that other students are also interested in. That is why we encourage students to use Piazza to ask questions in public that other students can see and answer and NOT ask them through E-mail in private UNLESS the question itself is about private matters to discuss with the professor.

10.4 Late Policy

Late submissions and late examination defence are not allowed. Exceptions may be made at a discretion of the professor only in force-majeure circumstances.

10.5 Incomplete

As with late exams, the grade *I* may be awarded only in special circumstances. The student must start discussion on getting the grade I with the instructors in advance and not during the last week before the final exams.

10.6 Academic Honesty

Plagiarism can be defined as an act or an example of copying or stealing someone elses words or ideas and appropriating them as ones own. The concept of plagiarism applies to all tasks and their components, including program code, abstracts, reports, graphs, statistical tables, etc.

In addition to being unethical, this indicates that the student has not studied the given material. Tasks written from somewhere for 5% or less will be assessed accordingly or will receive a 0 at the discretion of the teacher. If plagiarism is more than 5%, the case will be transferred to the AUCA Disciplinary Committee.

On this course team work is NOT encouraged. The same blocks of code or similar structural pieces in separate submissions will be considered as academic dishonesty and all parties will get zero for the task.

The following are examples of some common acts of plagiarism:

1. Representing the work of others as their own

2. Using other people's ideas or phrases without specifying the author

3. Copying code snippets, sentences, phrases, paragraphs or ideas from other people's works,published or unpublished, without referring to the author

4. Replacing selected words from a passage and using them as your own

5. Copying from any type of multimedia (graphics, audio, video, Internet streams), computer programs, graphs or diagrams from other people's works without representation of authorship

6. Buying work from a website or from another source and presenting it as your own work

Students are not recommended to memorize before exams, as this is a difficult and inefficient way to learn; and since practice exams consist of open questions designed to test a students analytical skills, memorization invariably leads to the fact that the answers are inappropriate and of poor quality.