COM 123, 410, Principles of Computing Systems, Computer Architecture

American University of Central Asia
Software Engineering Department

1 Course Information

Course Codes

COM-123 COM-410

Course IDs

5778
3268

Prerequisites

COM-117, Programming II
or
COM-310-311.1, Circuit Engineering
or
COM-223.1, Algorithms and Data Structures

Prerequisites for

COM-312, Neural Network and Deep Learning
COM-341, Operating Systems
COM-392, System Programming
COM-431, Senior Project Preparation I/II

Credits

6

Time and Place

Dmitrii Toksaitov

Lecture: Monday 12:45–14:00, CH
Lecture: Monday 14:10–15:25, CH
Lab: Wednesday 12:45–14:00, Lab 432
Lab: Wednesday 15:35–16:50, Lab 432
Lab: Friday 12:45–14:00, Lab 432
Lab: Friday 15:35–16:50, Lab 432

Pavel Ges

TBD

Course Materials

`https://github.com/auca/com.123-410`

2 Contact Information

Professor

Dmitrii Toksaitov
toksaitov_d@auca.kg
Pavel Ges
ges_p@auca.kg

TAs

Kamila Inchina
inchina_k@auca.kg

Office

AUCA, room 315

Office Hours

By appointment throughout the work week (write to your professor or TA to make an appointment)

3 Course Overview

The course introduces students to the topic of computer architecture and organization. Students will focus on studying the structure and design of modern central processing units. Students will learn the basics of the x86-64 and aarch64 instruction sets, the assembly languages for the platforms, and the representation of high-level language structures in the low-level languages.

At the end of the course student should be able to research, analyze, design, develop, and maintain software in low-level programming languages in accord to the goals of the AUCA Software Engineering Department and the 510300 IT competency standard (OK 17, 17, 115).

4 Topics

- Week 1–2: The modern computer architectures and organization (6 hours)
- Week 3–5: The x86-64 and ARM64 assembly languages (9 hours)
- Week 6–9: Representation of high-level language structures in low-level assembly languages (12 hours)
- Week 10–13: Acceleration with SIMD instructions (12 hours)

- Week 14–15: System emulation (6 hours)

5 Assignments and Exams

5.1 GitHub Checkpoints

Students will have to maintain a personal private GitHub repository with all their works shared with their instructor. Students have to periodically commit and push a specific number of lab and project solutions as told by the instructor. The instructor and TAs will regularly check the work and give points for the accomplished work.

5.2 Labs and Projects

Students will have a number of laboratory tasks. Students will have to defend their work to the instructor during midterm and final examination sessions.

At the end of the course, students will have to work on one major project. The work will require accelerating an image processing application by optimizing the hot path of a C program in x86-64 and ARM64 assembly.

6 Course Materials, Recordings and Screencasts

Students will find all the course materials on GitHub. We hope that 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 at any time MUST NOT be a reason not to attend classes. Active participation is necessary to succeed in this course.

7 Software

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

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

The compilers, assemblers and debuggers will be available on the remote course server.

8 Reading

- Computer Architecture: A Quantitative Approach, 6th Edition by David Patterson and John L. Hennessy (AUCA Library Call Number: QA76.9.A73 P377 2019, ISBN-13: 978-0128119051, ISBN-10: 0128119055)

- Assembly Language for x86 Processors, 6th or 7th Edition by Kip R. Irvine (AUCA Library Call Number: QA76.8.I77 2011, ISBN-13: 978-0136022121, ISBN-10: 013602212X)
- ARM 64-Bit Assembly Language by Larry D. Pyeatt and William Ughetta (ISBN-13: 978-0128192214, ISBN-10: 0128192216)

9 Grading

9.1 GitHub Checkpoints

Your instructor will announce a periodic review of your work. You will be awarded up to the following number of points for such checks.

- Labs (20%)
- Project (15%)

9.2 Exams

- Midterm Exam (30%)
- Final Exam (35%)

9.3 Totals

- 100% is formed from the GitHub submissions (35%) and the two exams (65%).

9.4 Scale

- [92%–100]%: A
- [85%–92)%: A-
- [80%–85)%: B+
- [75%–80)%: B
- [70%–75)%: B-
- [65%–70)%: C+
- [60%–65)%: C
- [55%–60)%: C-
- [50%–55)%: D+
- [45%–50)%: D
- [40%–45)%: D-
- Less than 40%: F

Please, note that requests to award a better grade if the number of points is close to such a grade will be ignored. For example, 91.99 is A-, NOT A. Likewise, requests to get extra assignments to increase the number of points will also be overlooked entirely.

10 Rules

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

10.1 Participation

Active work during the class may be awarded with extra points at the instructors discretion. Poor student performance during a class can lead to 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 lose points. Students absent without a good reason from such classes with graded work will also lose points unless it is force-majeure circumstances. TAs and instructors must be notified in advance about why a student is absent not to lose points.

10.2 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 the Canvas online discussion board to ask questions in public that other students can see and answer. We discourage students from asking questions through E-mail. If it is a private matter, write direct messages to TAs or your instructor there on Canvas too. We will not be answering most E-mail messages this semester (unless it is a severe emergency) to consolidate all the course correspondence in one place on our LMS (Learning Management System).

Do not post the complete source code for any task on the Canvas discussion board. You will get zero for that work for any such public post. Do not ask generic questions about your code to know why it does not work. Please spend some time thinking about your code, debugging it.

10.3 Late Policy

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 the online discussion board of the LMS (Learning Management System) that you use (e.g., AUCA e-Course System) to ask questions in public that other students can see and answer. We discourage students from asking questions through E-mail. If it is a private matter, write direct messages to TAs or your instructor through the LMS system (such as AUCA e-Course System) too. We will not be answering most E-mail messages this semester (unless it is a severe emergency) to consolidate all the course correspondence in one place.

Do not post the complete source code for any task on the LMS discussion board. You will get zero for that work for any such public post. Do not ask generic questions about your code to know why it does not work. Please spend some time thinking about your code, debugging it.

10.4 Exam Ceremonies

Students MUST follow exam ceremonies. It means they MUST prepare task list forms with all points appropriately calculated. They MUST submit them correctly. They must bring task list forms to the exam. Failure to do so will result in lost points. Throughout your career, you will have to work with various supporting documents (contracts, timesheets, etc.). It is a good idea to start learning to work with such documents accurately early. We will remove points for not following these rules or even refuse to accept your exam defense. We will also give zero for not following the strict exam timing rules.

10.5 Incomplete Grade

As with late exams, the grade *I* may be awarded only in exceptional circumstances. The student must start a 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, comments, documentation, abstracts, reports, graphs, statistical tables, etc.

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

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.

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

Students are not recommended to memorize lab and project code before exams, as this is a difficult and inefficient way to learn; and since practice exams may 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.