COM 122, 119: Introduction to Programming and Object-oriented Programming

American University of Central Asia
Software Engineering Department

1 Course Description

This two-semester course provides a foundation in programming, emphasizing structured, procedural, and object-oriented paradigms in C++. Offered by AUCA's Software Engineering Department, it is a prerequisite for many departmental courses. It is also suitable for students from other departments with no prior experience who want a general introduction to the field.

Students will learn essential concepts, including memory management, control flow, functions, procedural decomposition, objects, classes, encapsulation, inheritance, and polymorphism. The course emphasizes hands-on practice developing C++ applications while building proficiency with fundamental development tools such as code editors and IDEs, compilers, build systems, package managers, debuggers, and version control systems. By the end of the course, students will be equipped to write efficient, well-structured programs and to apply good programming practices consistently.

2 Course Details

Course Materials

`https://github.com/auca/com.122-119`

Course Codes

COM-122
COM-119

Course IDs

5682
4357

Prerequisites

COM-122:
None

COM-119:
COM-122, Introduction to Programming

Prerequisites for

The list below depends on the student's year of admission.

COM-122:
COM-119, Object-oriented Programming
COM-123, Principles of Computing Systems

COM-119:
COM-213, Database
COM-223, Algorithms and Data Structures
COM-229, Data Structures
COM-421, Software Engineering I
COM-431, Senior Thesis I
MAT-407, Numerical Methods
Various elective courses

Credits

6

Professors, Time, Place

Dmitrii Toksaitov
Lecture: Monday 14:10–15:25, CH 440
Lab: Wednesday 14:10–15:25, Lab 432

Adilet Abdykerimov
Lab: Wednesday 10:50–12:05, Lab G31
Lab: Wednesday 12:45–14:00, Lab G31
Lab: Wednesday 15:35–16:50, Lab G31
Lab: Friday 14:10–15:25, Lab G31

3 Contact Information

Professors

Dmitrii Toksaitov
toksaitov_d@auca.kg

Adilet Abdykerimov
abdykerimov_a@auca.kg

TAs

Luna Maltseva (Teacher Assistant)
md12366@auca.kg

Sofia Kan (Administrative Assistant)
ks12246@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 or TA to schedule an appointment.

4 Topics

4.1 Introduction to Programming

- Weeks 1–2: Introduction to the Process of Software Development (6 hours)
- Weeks 3–6: Selections (12 hours)
- Weeks 7–10: Loops (12 hours)
- Weeks 11–13: Functions (9 hours)
- Weeks 14–16: Single- and Multidimensional Arrays (9 hours)

4.2 Object-oriented Programming

- Weeks 1–3: Objects and Classes (9 hours)
- Weeks 4–5: Exception Handling (6 hours)
- Weeks 6–8: Encapsulation, Inheritance, and Polymorphism (9 hours)
- Weeks 9–10: Abstract Classes and Interfaces (6 hours)
- Weeks 11–12: GUI and Computer Graphics Basics (6 hours)
- Weeks 13–14: Templates (6 hours)
- Weeks 15–16: Working with I/O (6 hours)

5 Learning Outcomes

By the end of this course, students will:

1. **Acquire basic programming skills to write and test programs**:
   - Write programs using fundamental C++ language constructs
   - Apply debugging and testing techniques to ensure correctness
2. **Develop proficiency in modeling object-oriented systems**:
   - Represent software designs using object-oriented principles
   - Use appropriate modeling techniques to describe classes, objects, and relationships

3. **Evaluate and use appropriate tools and techniques for software development**:
    - Use code editors, compilers, build systems, debuggers, IDEs, and other development tools effectively
    - Employ best practices in version control and collaborative development
4. **Design, plan, and critically evaluate software solutions to problems**:
    - Work with user requirements and constraints
    - Assess possible solutions for correctness and maintainability
5. **Assess systems in terms of quality attributes and trade-offs**:
    - Use automated testing and code analysis tools to evaluate software quality
    - Weigh design decisions against project constraints and overall goals
6. **Reflect on and reason about information handling problems**:
    - Critically examine data structures and algorithms for effective information management
    - Consider privacy, data protection, and ethical implications of software solutions
7. **Understand and apply ethical principles and professional standards**:
    - Demonstrate awareness of legal and ethical issues in computing
    - Practice professional conduct and responsibility in the design and implementation of software

6 Assignments

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 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 OJ Problems and Projects

Students will be required to develop a course project that demonstrates real-world applications. Depending on the course offering, they may also be assigned problems from an online judge (OJ) platform, such as Codeforces, to further enhance their problem-solving skills and prepare for programming contests or job interviews. They must present and defend their work to the instructor during the midterm or final examination period.

7 Course Materials, Recordings, and Screencasts

All course materials are available on GitHub at `https://github.com/auca/com.122-119`. 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.122-119`.

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-122-zoom.auca.space` or `http://com-119-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 request permission from the instructor in advance. Please send a brief email explaining why you need to attend remotely (e.g., illness, personal reasons) and provide appropriate documentation to support your request, either immediately or as a follow-up in the same email thread. If documentation is not provided within a reasonable timeframe, you will be marked absent without excuse and may receive a grade of X if you accumulate three or more unexcused absences. Given the substantial number of such requests we receive, do not expect a reply to your email. Regardless, you must still submit all required documentation as soon as possible in the same email thread. AUAF students who are outside the country and can only participate online do not need to request this permission.

8 Software

Students are advised to install the following software on their computers at the start of the course. Additional installations may be required later.

- Git: `https://git-scm.com/downloads`
- CLion: `https://www.jetbrains.com/clion/download`

Generally, we select tools and design tasks so you can use any modern operating system (Windows, macOS, or Linux) to complete the course. However, we officially support only the operating systems used on the lab computers (Windows 10 or 11) and automatic graders (Ubuntu 24.04). If you cannot complete an exercise on your own computer, ask the teaching staff for help troubleshooting. If the issue cannot be resolved, please use the lab machines.

9 Hardware

There are no specific hardware requirements, as you can use lab machines where we guarantee you will be able to complete all the tasks of our course. However, we recommend a machine with more memory (rather than less), high-speed disk I/O, and a GPU compatible with current graphics APIs.

10 Reading

1. Introduction to Programming with C++, 3rd edition by Y. Daniel Liang (AUCA Library Call Number: QA76.73.C153 L53 2014, ISBN: 978-0273793243)

10.1 Supplemental Reading

1. The C++ Programming Language, Fourth Edition by Bjarne Stroustrup (AUCA Library Call Number: QA76.73.C153 S77 2013, ISBN: 978-0275967307)
2. A Tour of C++, Third Edition by Bjarne Stroustrup (AUCA Library Call Number: QA76.73.C153 2023, ISBN: 978-0136816485)

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 is accurate until all tasks are in the system and your instructor has asked you to review your scores and grades.

One common mistake students make is assuming that attendance points are part of the grade and will be summed at the end—they will not. These points are used only to help us decide on X grades in the middle of the semester and will be removed from the totals in Moodle at the end of the semester.

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 Submissions and Class Work (16%)
- Online Judge Problems (16%, COM-122 only)
- Course Project (13% for COM-122, 29% for COM-119)

For COM-119, the Course Project consists of lab-like parts completed with the instructor during the second half of the semester, worth 16% in total, and an independent part, worth 13%. Together they form the 29% you can get for the Course

Project. Do note, the price for the labs in the first part of the semester is higher as tasks are more complicated. In total they lab submission unrelated to the Course Project still costs 16%.

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 (30%)

11.3 Bonus Opportunities

You may earn up to the number of bonus points listed below to recover lost points from other tasks by attending Extra Lecture Sessions or visiting the Writing and Academic Resource Center (WARC). One point is awarded for each visit.

- Extra Lecture Sessions or WARC Bonus (+5%)

11.4 Totals

100% is formed from the Moodle/e-Course Checkpoints (45%) and the exam sessions (55%). You can earn up to 5% as a bonus by attending WARC or Extra Lecture Sessions conducted by your instructors.

11.5 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.

You can attend WARC consultation sessions conducted by students or Extra Lecture Sessions led by instructors. For each session you attend, you will earn one extra point (up to a maximum of five points in total). We will collect your Extra Lecture and WARC attendance reports at the end of the course and add these points to your course total when calculating your final grade.

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 connection, you must notify the instructors at least 24 hours before the deadline. Failure to do so will result in no extension being granted. Requests made on the day of the deadline will be treated as procrastination and will not be considered. No student will be granted more than one emergency extension throughout the course.

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

12.4 Exam and Task Submission Ceremonies

Students must follow all exam and task submission protocols. This means strictly adhering to all rules specified by instructors, whether in written or verbal form. Failure to do so will result in a loss of points.

Throughout your career, you will work with various supporting documents such as contracts and timesheets. Beyond programming, you will also need to use project management systems, application life cycle management tools, version control, and continuous integration and deployment pipelines. You will manage complex configurations, collaborate with colleagues by following specific methodologies, and share code through pull requests or other means. In all these processes, proper etiquette and attention to detail are essential.

In our courses, we aim to help you build these skills. We will deduct points for not following submission rules or protocols, and we may even refuse to accept your exam defense or submitted tasks. We will also award zero points for missing task deadlines or violating strict exam timing rules, such as arriving late or failing to submit exam documents on time.

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 exam; requests made the day before the final exam or later 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, the student who plagiarizes and the student whose work was copied, will face equal consequences, as every student is responsible for safeguarding their assignments, code, and related materials to prevent unauthorized use. This policy reflects real-world expectations in software engineering, where failing to protect intellectual property can result in termination, reputational damage, compromise the safety of customers and users, and even lead to lawsuits in a professional environment.

All work in this course must be completed individually. Group work, collaboration with peers, or seeking assistance from other students is strictly prohibited. At the introductory level, relying on classmates often leads to the spread of misunderstandings and increases the risk of academic dishonesty. Students may seek help only from instructors, teaching assistants (TAs), or WARC tutors, ensuring that guidance comes from qualified sources and that each student develops their own skills and understanding.

The use of artificial intelligence, including but not limited to generative AI tools, to complete any assignments, projects, or exams, either off-site or on-site, is strictly prohibited. If there is suspicion that a student has used AI assistance in their work, the student will be required to perform a similar task and answer relevant questions in a supervised setting with the course instructors. Failure to satisfactorily complete this task or to answer questions convincingly will lead to the conclusion that the work was AI-generated. In such cases, the Academic Honesty policies outlined previously will be enforced, including potential disciplinary actions.

Finally, this course adheres to all global AUCA 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 AUCA 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 AUCA Writing and Academic Resource Center (WARC). Tutoring is available both in person and online. Additional resources can be found on the WARC webpage. Please note, however, that team or group work with other students is not permitted for assignments in this course.

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.