

COM 118, 122, 119: Structured Programming, Introduction to Programming, Object-oriented Programming

American University of Central Asia
Software Engineering Department

1 Course Information

Course Codes

COM-118
COM-122
COM-119

Course IDs

4322
5682
4357

Prerequisite for

COM-118, COM-122:

COM-119, Object-oriented Programming

COM-119:

COM-223, Algorithms and Data Structures

Credits

6

Professors, Time, Place

Dmitriy Shostak

Lecture: Monday 14:10–15:25, CH

Dmitriy Shostak

Lab: Wednesday 10:50–12:05, Lab G31

Lab: Wednesday 12:45–14:00, Lab G31

Lab: Wednesday 14:10–15:25, Lab G31

Pavel Ges

Lab: Tuesday 14:10–15:25, Lab 433

Lab: Wednesday 12:45–14:00, Lab 233

Dmitrii Toksaitov

Lab: Wednesday 12:45–14:00, Lab 432

Lab: Wednesday 14:10–15:25, Lab 432

Lab: Wednesday 15:35–16:50, Lab 432

2 Contact Information

Professors

Dmitriy Shostak

shostak_d@auca.kg

Pavel Ges

ges_pa@auca.kg

Dmitrii Toksaitov

toksaitov_d@auca.kg

Office

AUCA, room 315

Office Hours

Individually decided by each instructor

3 Course Overview

This course helps to equip students with essential skills needed for structured and object-oriented programming. At the completion of the course, students should understand fundamental programming concepts such as memory management, flow control, methods, procedural decomposition, objects, classes, inheritance, and polymorphism; be able to write simple applications using most of the capabilities of the Java programming language and apply principles of good programming practices throughout the process.

At the end of the course, the student should be able to research, analyze, design, develop, and maintain functioning software systems according to the goals of the AUCA Software Engineering Department and the 510300 IT competency standard.

4 Topics Covered

Structured Programming, Introduction to Programming

- Week 1: Introduction to the Process of Software Development (3 hours)
- Week 2: Working with Memory (3 hours)
- Week 3–5: Selections (9 hours)
- Week 6–9: Loops (12 hours)
- Week 10–12: Methods, Procedural Decomposition (9 hours)

- Week 13–14: Single- and Multidimensional Arrays (6 hours)

Object-oriented Programming

- Week 1–3: Objects and Classes (9 hours)
- Week 4–6: Inheritance and Polymorphism (9 hours)
- Week 7–8: Abstract Classes and Interfaces (6 hours)
- Week 9–10: Exception Handling (6 hours)
- Week 11–12: GUI and Computer Graphics Basics (6 hours)
- Week 13–14: Generics and Container Classes (6 hours)
- Week 15–16: Working with I/O (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 online judge solutions (or project parts) as told by the instructor. The instructor of lab classes will regularly check the work and give points for the accomplished work.

Students must create a separate GitHub account for the course with a login name '`<last-name-in-lowercase>-<first-letter-of-the-first-name>-auca-2022`'. For example: '`shostak-d-auca-2022`'

The GitHub repository must be named '`intro-to-programming-2022`'.

The GitHub repository structure must be the following:

```
lab-01
  Problem01.java
  Problem02.java
  ...
  Problem10.java
lab-02
  Problem01.java
  Problem02.java
  ...
  Problem10.java
  ...
oj-midterm
  Problem01.java
```

```

        Problem02.java
        ...
    oj-final
        Problem01.java
        Problem02.java
        ...
    project-01
        Problem01.java
        Problem02.java
        ...
    project-02
        Problem01.java
        Problem02.java
        ...

```

Each directory must be an IntelliJ IDEA's module.

The repository must include a *.gitignore* file with the following line:

```
out/
```

Points may be removed for not following these organizational rules or any default IDE style formatting.

5.2 Labs, Online Judges Tasks, Projects

Students will have eight laboratory tasks, get a number of problems from online judge systems, and have to finish two projects. Students will have to defend their work to the instructor during midterm, final, and project defence examination sessions.

6 Software

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

- Adoptium Temurin JDK 17: <https://adoptium.net/en-GB/>
- IntelliJ IDEA **Community Edition**: <https://www.jetbrains.com/idea/download>
- Git VCS: <https://git-scm.com>

All students must configure *Git* once on every new machine that they use with the following commands (with their name and email):

```
git config --global user.name "John Doe"
git config --global user.email john_d@auca.kg
```

7 Reading

Introduction to Java Programming and Data Structures, Comprehensive Version, 12'th Edition by Y. Daniel Liang (ISBN-13: 978-0136520238, ISBN-10: 0136520235)

8 Grading

8.1 GitHub Checkpoints

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

- Labs (10%)
- Online Judge Problems (10%)
- Project #1 (4%)
- Project #2 (4%)

8.2 Interview

- Labs 1–4 (4% + 4% + 4% + 4%)
- Online Judge Problems (10%)

Midterm Exam (Labs + Online Judge Problems)

- Labs 5–8 (4% + 4% + 4% + 4%)
- Online Judge Problems (10%)

Final Exam (Labs + Online Judge Problems)

- Project #1 (10%)
- Project #2 (10%)

Project Defense (Both projects will be checked during or after the Final Exam)

8.3 Total

- 100% is formed from the GitHub submissions (28%) and the interview exam sessions (72%).

8.4 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

9 Rules

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

9.1 Attendance

Strict class attendance is mandatory. Students will lose one point from their grades for every missed class. Students that are late to the attendance check are considered not present.

9.2 Late Policy

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

9.3 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.

9.4 Incomplete

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.

9.5 Academic Honesty

Plagiarism can be defined as “an act or an example of copying or stealing someone else’s words or ideas and appropriating them as one’s 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 student’s analytical skills, memorization invariably leads to the fact that the answers are inappropriate and of poor quality.