	Engineering Faculty **Computer Engineering Department**	**Syllabus**

COURSE INFORMATION

Academic Year / Semester		2025–2026 / Spring
Course Code, Name and Credits		**LCEN462 Graduation Project (Capstone Design) — 1 Credit / 8 ECTS (AKTS) — Section 6**
Lecture and Project Schedule	**Seminar**	Graduation Project seminars; schedule announced weekly
	Project Work	Weekly team progress meetings, supervised project work, reports, demonstrations and feedback
Course Type		Compulsory
Prerequisite		Senior standing; completion of core Computer Engineering courses. Recommended background: calculus, linear algebra, statistics and probability, programming, data structures and algorithms.

ACADEMIC PERSONNEL

Title / First and Last Name	Prof. Dr. Haluk Gümüşkaya
Room and Phone	Room 338
E-Mail Address / Web Page	**E-mail:** halukgumuskaya@arel.edu.tr; haluk@gumuskaya.com **Web:** www.gumuskaya.com | Course page: Google Classroom (class code: 4326ucn2)
Office Hours and Assistant	TBA; team appointments and progress meetings will be announced weekly

COURSE DESCRIPTION

The Graduation Project (Capstone Design) course is a culminating, industry-oriented design experience in which senior Computer Engineering students integrate and apply knowledge acquired throughout the undergraduate program to develop a complete, working engineering system under faculty supervision.

Students work individually or in teams—normally about two students—to define a real problem, review the state of the art, specify requirements, design an architecture, implement and integrate hardware and/or software, test and benchmark the result, and deploy or publicly demonstrate a working prototype or validated simulation.

Projects may involve artificial intelligence and machine learning, computer vision and embedded systems, cybersecurity, distributed and cloud systems, data engineering, IoT and smart systems, robotics, human–computer interaction, high-performance computing, edge AI, generative AI/LLM applications, or another approved field.

Every project must include a well-defined problem statement, a clear technical novelty or engineering contribution, measurable performance metrics, and a working prototype or validated simulation. Relevant economic, environmental, ethical, security, privacy, scalability, reliability, manufacturability and regulatory constraints must also be considered.

	Engineering Faculty **Computer Engineering Department**	**Syllabus**

COURSE OBJECTIVES

By the end of this course, students will be able to:

1. Design a complete computer engineering system under realistic constraints.
2. Apply theoretical knowledge to practical, real-world problems.
3. Conduct literature research and evaluate state-of-the-art solutions.
4. Develop and integrate hardware and/or software components.
5. Perform experimental validation and data-driven performance analysis.
6. Work effectively in multidisciplinary teams.
7. Communicate technical results in written and oral formats.
8. Demonstrate professional ethics and engineering responsibility.

Professional skills developed: project management, technical leadership, research methodology, technical writing, presentation, innovation and entrepreneurship.

COURSE TEXTBOOK AND OTHER MATERIALS

Main Materials	• No single textbook is prescribed; current, project-specific sources will be used. • Graduation Project course overview, templates, rubrics and announcements distributed through Google Classroom. • Peer-reviewed literature and state-of-the-art technical papers relevant to the approved project. • Standards, datasets, technical manuals, vendor documentation and software/hardware platform documentation required by the project. • Final technical report in the announced thesis or academic-paper format, supported by proper references and academic-integrity practices.
Tools	Project-appropriate version control and issue tracking; CI/CD; AI-assisted development; simulation and testing environments; cloud and container platforms; data/ML tools; and embedded hardware development kits. Exact tools depend on the approved project.

WEEKLY PROJECT SCHEDULE (Tentative and updated weekly)

The schedule integrates seminars, project reviews, progress meetings and independent team work. All dates and deliverables are tentative; approved updates will be announced in Google Classroom. Teams should maintain a continuously updated project plan, repository and academic-format report.

W/D	Seminar / Review Topics	Weekly Project Activities and Deliverables
1 **18/02**	Course overview; expectations; engineering lifecycle; project scope and types; professional and ethical responsibilities.	Join Google Classroom; complete background/expectations activity; form teams (normally ~2 students); identify candidate real problems, clients and data.
2 **25/02**	Problem definition and needs analysis; the 4Rs: real client, real problem, real data and real deadline; feasibility and engineering constraints.	Select a project direction; define stakeholders, need, preliminary scope, success criteria and measurable performance metrics.
3 **04/03**	Project proposal and plan; teamwork roles; schedule, resources, risks, communication and version-control workflow.	Submit/present Proposal & Project Plan; establish repository, issue tracker, meeting log and continuously updated report.

W/D	Seminar / Review Topics	Weekly Project Activities and Deliverables
4 **11/03**	Literature review, systematic search, state-of-the-art evaluation, related work and gap analysis; academic integrity and citation practice.	Build the literature matrix; state the technical novelty or engineering contribution; begin the related-work section.
5 **18/03**	Requirements specification and feasibility analysis; functional/nonfunctional requirements; constraints, ethics, security, privacy and sustainability.	Progress Meeting 1: review problem, requirements, risks and evidence; revise the project plan and requirements specification.
6 **25/03**	System architecture and high-level design; hardware/software partitioning; interface, data and deployment design; technology selection.	Prepare architecture views and rationale; define components, interfaces and integration strategy; prototype critical risks.
7 **01/04**	Verification and validation strategy; experiment/test planning; datasets, baselines, metrics, traceability and acceptance criteria.	Architecture/validation review; complete literature review and architecture report; create test and experiment plan.
8 **08/04**	Implementation Iteration 1; incremental integration; reproducibility, configuration management and evidence-based progress reporting.	Progress Meeting 2 and prototype demo; record responsibilities, technical decisions, issues, results and next actions.
9 **15/04**	Midterm examinations period. No separate written LCEN462 midterm is planned; supervised project work continues.	Consolidate proposal, requirements, related work, architecture and validation plan; continue implementation and weekly report updates.
10 **22/04**	Implementation Iteration 2; version control, automated builds/tests, CI/CD and AI-assisted engineering workflows used responsibly.	Progress Meeting 3 and integration demo; provide repository evidence and updated technical report; address feedback.
11 **29/04**	Technical contribution and system integration; data/model/component quality; interface testing and defect/risk management.	Progress Meeting 4; demonstrate an integrated increment; document technical contribution, limitations and remaining risks.
12 **06/05**	Experiments, results and error analysis; performance benchmarking, comparison with baselines and reproducibility.	Run planned experiments/tests; organize quantitative and qualitative results; analyze errors, threats to validity and limitations.
13 **13/05**	Implementation Iteration 3; reliability, scalability, security/privacy, usability and other project-specific engineering constraints.	Progress Meeting 5 and near-final demo; close critical defects; update results, discussion and constraint analysis.

W/D	Seminar / Review Topics	Weekly Project Activities and Deliverables
14 20/05	Final system integration and acceptance; technical documentation; academic paper/thesis structure; poster and public-demo preparation.	Freeze the main feature set; verify traceability and acceptance criteria; circulate the draft report and presentation for feedback.
15 27/05	Eid al-Adha Holiday.	Continue final report, poster/video, repository documentation and demo preparation as appropriate; monitor official announcements.
16 03/06	Project completion review; public demo/poster rehearsal; professional presentation and technical-defense preparation.	Submit the working prototype or validated simulation and implementation artifacts; complete the report and presentation package before the announced deadline.
	Final Examinations / Final Makeup Examinations Period	Final technical report, public presentation and defense in the announced slot; makeup arrangements, if any, follow University and course announcements.

ASSESSMENT AND GRADING

Assessment is based on the following project work products. ADC means Attendance/Discussion/Contribution and includes weekly communication, meetings, progress reports, demonstrations, timely feedback and evidence of each member's contribution.

Assessment Component	Evidence and Focus	Weight
Proposal & Project Plan	Problem, need, scope, stakeholders, feasibility, constraints, metrics, plan, roles and risks.	10%
Literature Review & Architecture Report	State-of-the-art research, gap/contribution, requirements, high-level design, architecture and validation plan.	15%
Communication, Progress (ADC)	Consistent weekly progress, attendance/discussion/contribution, teamwork, reports, demos, feedback and individual contribution.	15%
Implementation & Technical Contribution	Working hardware/software, integration quality, technical/scientific contribution, testing, reproducibility and repository evidence.	25%
Final Report (Thesis Format)	Clear academic-format report with related work, method, implementation, experiments, results, error/limitation analysis and references.	20%
Final Presentation & Defense	Professional public presentation/demo, poster or supporting media, technical explanation and answers during the defense.	15%
TOTAL		**100%**

REQUIRED DELIVERABLES AND PROJECT EXPECTATIONS

- Project proposal and project plan
- Requirements specification
- Related-work / literature-review and gap analysis
- System architecture document and validation plan
- Implementation artifacts (code, data/models and/or hardware) with repository documentation
- Experimental/test results and performance analysis
- Final technical report in the announced thesis or academic-paper format
- Poster and public demo; final presentation and technical defense

Important: The course is conducted as a supervised, team-oriented engineering project. Teams must make real, measurable progress, keep their project artifacts current, cite sources properly, disclose and use AI-assisted tools responsibly, and be able to explain and defend every submitted work product. Approved updates are announced through Google Classroom.