CS 256 — Topics in Artificial Intelligence

Syllabus · Fall 2026 · San José State University

Course Information

Course	CS 256 — Topics in Artificial Intelligence (3 units)
Semester	Fall 2026
Section	02
Meeting days/ times	Tuesdays & Thursdays, 6:00 – 7:15 PM
Location	MacQuarrie Hall (MH) 225
Modality	**In person.** All class meetings, in-class exercises, and project demonstrations take place on campus.
Instruction dates	August 19 – December 7, 2026
Final exam	During the SJSU final examination period, December 9–15, 2026 — [TBD: exact date/time per the official final exam schedule]

Instructor

Instructor	Amer Rez
Email	amer.rez@sjsu.edu
Office	Duncan Hall (DH) 282
In-person office hours	Thursdays after class, from 7:15 PM, **every other week**, beginning Week 3 (Sept 3), in DH 282. No fixed end time — I stay as long as students need.
Virtual office hours	Mondays, 5:30 PM, weekly. [TBD: duration + meeting link]
By appointment	Additional meetings available by appointment at mutually convenient times.

Scheduling note: Office hours are not held on days the campus is closed — Monday, Sept 7 (Labor Day) and Thursday, Nov 26 (Thanksgiving).

Course Description and Requisites

This section of CS 256 covers **modern generative AI and large language models (LLMs)**. We work from the ground up: how text becomes tokens and embeddings, how transformers and attention work, how generative models produce text, and how these models are applied, optimized, and extended. The second half moves into applied systems — prompt engineering, AI agents that call tools, retrieval augmented generation (RAG), and the performance optimizations that make inference practical.

The course is **hands-on**. You will load and run real models locally, measure their performance, and build two working systems from scratch across two multi-milestone projects.

Prerequisites

CS 156 and Graduate standing. Allowed declared majors: Computer Science, Bioinformatics, Data Science. Or instructor consent.

You should be comfortable with Python programming and have prior exposure to machine learning fundamentals.

Course Goals

By taking this course, you will:

1. Understand how modern language models represent and generate text — from tokenization and embeddings through attention, transformer architecture, and the difference between encoder-only and decoder-only (generative) models.
2. Be able to load, run, and evaluate open language models on your own machine, and measure real inference performance.
3. Understand **inference-time optimizations** — KV caching and grouped-query attention — and be able to measure the speed and memory they actually save.
4. Understand **advanced architectural optimizations** used in modern GenAI systems — sparse attention, FlashAttention (tiling, online softmax, kernel fusion), and rotary positional embeddings (RoPE) — and explain what each one trades off.
5. Understand **what an AI agent is and how one actually works** — the agent loop, tool-calling protocols, action parsing, tool dispatch, and the different agent styles (ReAct, plan-and-execute, graph-based).
6. Understand **agent optimization techniques, especially memory handling** — conversation buffers, windowed buffers, summarization memory, and prompt chaining — and the tradeoffs each makes as conversations grow.
7. Be able to design effective prompts and understand the mechanics behind prompt-based techniques.
8. Build working AI systems — a tool-calling agent and a retrieval-augmented (RAG) system — and be able to explain your own implementation in depth.

Course Learning Outcomes (CLOs)

Upon successful completion of this course, students will be able to:

- **CLO 1** — Explain how text is tokenized and embedded, and describe how transformer attention produces contextual representations.
- **CLO 2** — Explain how generative models produce text **autoregressively**, including the distinction between encoder-only (representation) and decoder-only (generative) architectures, the role and limits of the context window, and how sampling strategies such as greedy decoding and temperature affect the output.
- **CLO 3** — Describe how language models are **pre-trained and fine-tuned**, and explain what distinguishes a base model from an instruct or chat model.
- **CLO 4** — Load an open language model, run inference, and measure performance characteristics such as latency and tokens per second.
- **CLO 5** — Apply language models to downstream tasks including **text classification and clustering**, and justify choosing between a fine-tuned encoder model and a generative LLM for a given task.
- **CLO 6** — Apply prompt engineering techniques and explain the mechanisms behind chaining, memory, and structured output.
- **CLO 7** — Explain and quantify inference-time optimizations including KV caching and grouped-query attention, and describe architectural optimizations such as sparse attention, FlashAttention, and rotary positional embeddings.
- **CLO 8** — Implement an AI agent that reasons over a tool-calling protocol, including action parsing, tool dispatch, and conversation memory; and compare agent styles (ReAct, plan-and-execute, graph-based) and their tradeoffs.
- **CLO 9** — Explain and compare agent memory strategies — full conversation buffer, windowed buffer, and summarization memory — including how each one affects context growth, cost, and answer reliability.
- **CLO 10** — Implement a retrieval augmented generation system using embeddings and similarity search, and explain why semantic retrieval outperforms keyword matching.
- **CLO 11** — Defend their own implementation work orally, demonstrating individual understanding of the code they submitted.

Course Materials

Highly recommended textbook

Hands-On Large Language Models — Jay Alammar & Maarten Grootendorst (O'Reilly). Used throughout the course.

Recommended supplementary reading (free, online)

How To Scale Your Model — Austin et al., Google DeepMind. https://jax-ml.github.io/scaling-book/ Useful for the performance and optimization unit.

Course platforms

- **Canvas** — slides, assignments, due dates, announcements, grades
- **GitHub** — starter code and course resources: **[insert GitHub link here]**

Lecture slides

Slides are posted on Canvas, organized by topic. Slides are the primary reference for lecture material and often contain assignment instructions.

Course Requirements and Assignments

University policy S16-9 requires that students spend a **minimum of 45 hours per unit of credit** over the semester, including class time. For this 3-unit course, expect approximately **135 hours total**, or roughly **9 hours per week** including class meetings, reading, and assignment work.

Homework (10% total)

Two programming assignments, submitted for automated grading. Both are hands-on and build directly toward the projects.

- **Homework 1 (6%)** — Load a language model, run inference, and measure inference time. This assignment establishes the model-loading workflow used throughout the rest of the course.
- **Homework 4 (4%)** — Inference-time optimizations. Measure the speedup from KV caching, and compute the memory savings from grouped-query attention.

Projects (45% total)

Two multi-milestone projects, each concluding with a **required in-person demonstration**.

Project 1 — AI Agent (25% of course grade) Build a tool-calling AI agent from scratch.

Milestone	% of project	% of course
M1 — Implement the agent's tools	40%	10%
M2 — Implement the action protocol and agent reasoning loop	40%	10%
M3 — Run against a real model + **in-person demonstration**	20%	5%

Project 2 — Web Retrieval and RAG (20% of course grade) Extend the agent with retrieval capability.

Milestone	% of project	% of course
M1 — Web content tool with relevance filtering	40%	8%
M2 — Embedding-based retrieval (RAG)	40%	8%
Demo — **In-person demonstration**	20%	4%

In-person project demonstrations

Each project concludes with a scheduled in-person demonstration during class. You will run your own code and answer questions about your implementation. **Demonstrations are individual**, and you must be able to explain any part of the code you submitted. Demonstrations cannot be completed remotely or made up without prior arrangement.

Examinations (43% total)

- **Midterm exam (18%)** — theory, in class. [TBD: date]
- **Final exam (25%)** — theory, during the university final exam period, December 9–15, 2026. [TBD: exact date/time]

Participation (2%)

Participation is earned through **two in-class exercises** delivered via Canvas — one theory exercise and one problem-solving exercise. Each is worth **1% of the course grade**. These take place during class meetings; dates will be announced in advance on Canvas.

Attendance itself is not graded. The participation grade is based solely on completion of these two graded in-class exercises.

Late work policy

Late submissions are penalized **5% per day**, counted from the posted due date, for a maximum of **6 days (30% total deduction)**. Work submitted more than 6 days late receives no credit. Extensions for documented emergencies are available — contact the instructor as early as possible. **In-person project demonstrations cannot be submitted late** and must be rescheduled in advance if you have a conflict.

Academic integrity

All project work is verified through individual in-person demonstration, where you must explain your own implementation.

Grading Information

Grade breakdown

Component	Weight
Theory midterm	18%
Theory final	25%
Project 1 — AI Agent	**25%**
· M1 tools	10%
· M2 protocol + agent loop	10%
· M3 in-person demo	5%
Project 2 — Web Retrieval & RAG	**20%**
· M1 web tool	8%
· M2 RAG retrieval	8%
· In-person demo	4%
Homework 1	6%
Homework 4	4%
Participation (2 in-class exercises)	2%
Total	**100%**

Grading scale

Grade	Range		Grade	Range
A plus	97–100%		C plus	77–79%
A	93–96%		C	73–76%
A minus	90–92%		C minus	70–72%
B plus	87–89%		D plus	67–69%
B	83–86%		D	63–66%
B minus	80–82%		D minus	60–62%
			F	Below 60%

Graduate students should note that a grade of C minus or lower may not satisfy degree requirements. Consult your program advisor.

Feedback timeline

Graded work will be returned within a reasonable timeframe. Programming assignments receive automated feedback on submission; exams and demonstrations are typically graded within one week.

Classroom Protocols

This is an in-person course. Class meets Tuesdays and Thursdays, 6:00–7:15 PM in MH 225. Lectures, in-class exercises, and project demonstrations all take place in the classroom.

- **Bring your laptop.** Many sessions include live coding and hands-on work.
- **In-class exercises** are delivered through Canvas during class and count toward your participation grade.
- **Project demonstrations** are scheduled in-person sessions. Plan to attend.
- Please keep devices used for non-course purposes to a minimum out of respect for those around you.
- Recording of class sessions requires prior instructor permission (see University Policies below).

Dropping and Adding

Students are responsible for understanding the policies and deadlines for add/drop and grade forgiveness. Refer to the current Academic Calendar and the Registrar page.

Technology Requirements

You will run language models locally on your own machine. Please ensure you have:

- **A laptop** you can bring to class
- **Python 3.11** — specifically 3.11.9 (the final 3.11 release)
- **At least 8 GB RAM**; 16 GB recommended for the largest models used
- **Roughly 10 GB free disk space** for downloaded models
- **Reliable internet** for model downloads (models are cached after first use)

Setup guides are provided with the assignments. No GPU is required — all assignments run on CPU.

If you do not have access to a suitable computer, contact the instructor early in the semester so we can find an alternative.

University Policies

University-wide policy information relevant to all courses — academic integrity, accommodations, religious holidays, recording, timely feedback, and student workload — is available on the Syllabus Information web page. Please visit that page to review these policies and resources.

Academic Integrity (Policy F15-7) — Your commitment as a student to learning is evidenced by your enrollment at San José State University. The University Academic Integrity Policy requires you to be honest in all your academic course work. Faculty members are required to report all infractions to the Student Conduct and Ethical Development office.

Course Schedule

The schedule is subject to change with fair notice. Any changes will be announced in class and posted on Canvas.

Semester dates: First day of instruction Wednesday, August 19, 2026 · Last day of instruction Monday, December 7, 2026 · Final examination period December 9–15, 2026.

Campus closures affecting this course: Thursday, November 26 — Thanksgiving, campus closed. This class meeting is lost and is not made up; Week 15 therefore has a single meeting (Tuesday, Nov 24), and course content is planned around 30 class meetings rather than 31. Monday, September 7 (Labor Day) affects the Monday virtual office hour only.

Week	Dates (Tue / Thu)	Topic	Assignments & Milestones
1	Aug 20 (Thu only)	Course intro; NLP & ML core concepts	
2	Aug 25, 27	Language AI, tokenization, embeddings, transformers, attention	
3	Sep 1, 3	Generative AI	**HW 1 assigned**
4	Sep 8, 10	Generative AI	
5	Sep 15, 17	Generative AI	**HW 1 due** · **Project 1 assigned**
6	Sep 22, 24	GenAI applications: classification, clustering, advanced generation	
7	Sep 29, Oct 1	GenAI applications (cont.)	**Project 1 — M1 due**
8	Oct 6, 8	Prompt engineering	**Midterm exam** [TBD: date — suggested Thu Oct 8]
9	Oct 13, 15	Prompt engineering (cont.)	
10	Oct 20, 22	Performance & optimization of GenAI	**HW 4 assigned Tue Oct 20** · **Project 1 — M2 due Thu Oct 22**
11	Oct 27, 29	Performance & optimization (cont.) · **Project 1 demos**	**Project 1 — M3 in-person demo**
12	Nov 3, 5	Performance wrap-up (Tue) · **RAG begins (Thu)**	**HW 4 due Tue Nov 3** · **Project 2 assigned**
13	Nov 10, 12	RAG (cont.)	**Project 2 — M1 due Tue Nov 10**
14	Nov 17, 19	RAG project	**Project 2 — M2 due Tue Nov 17**
15	Nov 24 (Tue only)	**Project 2 demos** — *no class Thu Nov 26 (Thanksgiving)*	
16	Dec 1, 3	**Project 2 demos** · final review	
Finals	Dec 9–15	**Final examination**	[TBD: exact date/time]

This syllabus is subject to change. Course content, materials, dates, and deadlines may be adjusted as needed during the semester. Any changes will be announced in class and posted to Canvas.