# CSCE 631-D: Intelligent Agents Computational Game Solving

Syllabus — Fall 2026

Department of Computer Science and Engineering
Texas A&M University — College Station

## 1 Course Information

**Course:** CSCE 631-D — Intelligent Agents: Computational Game Solving
**Section:** Online / asynchronous (lecture recordings; no synchronous meetings). Both sections use identical materials and identical scoring.
**Term:** Fall 2026 (15 weeks)
**Credit Hours:** 3
**Meeting format:** Online / Asynchronous

### 1.1 Instructor Details

**Instructor:** Dr. Alan Kuhnle
**Office:** 421 Peterson Building (PETR)
**E-Mail:** kuhnle@tamu.edu
**Office Hours:** Mondays 9:00–9:50, Thursdays 3:00–3:50

## 2 Course Description

This course studies the theory and practice of intelligent agents that make strategic decisions. We cover the classical foundations — normal-form games, equilibrium computation, regret minimization, extensive-form games, and counterfactual regret minimization — and connect each topic to the design and evaluation of modern LLM-based agents. Students will analyze game-solving algorithms, reproduce results from landmark papers, build LLM agent controllers, and investigate multi-agent interactions through a course project.

Prior experience with machine learning is helpful but not required; mathematical maturity (proofs, probability, optimization) is expected.

**Design principle.** Every module has a *theory core* (classical game theory) and an *agent thread* (modern LLM agent application). The agent thread motivates the theory from Week 1, not Week 12. Students program with LLMs starting in PA1 — TAMU API setup happens in Week 1.

**Assessment philosophy.** No exams. This is a graduate course — assessment is through paper reproduction (programming assignments) and creative research application (course project). Every PA is grounded in a specific published paper.

### 2.1 Prerequisites

CSCE 420 or CSCE 625.

## 3 Course Learning Outcomes

Upon completion of this course, students should be able to:

1. Understand fundamental concepts and techniques used in Multi-Agent Systems, including the elements of game theory.
2. Analyze strategic interactions among rational agents in diverse settings such as auctions, negotiations, and resource allocation.
3. Design algorithms that optimize outcomes in strategic environments and account for the incentives and motivations of rational agents.
4. Design and evaluate LLM-based agents using game-theoretic principles, including action abstraction, multi-agent coordination, and intervention strategies.
5. Reproduce and critically analyze results from published research in computational game theory and LLM agent systems.
6. Communicate effectively about Multi-Agent Systems concepts and ideas.

## 4 Course Structure

- **Lectures:** Delivered in person and recorded; the asynchronous section receives the same recordings. Both sections share identical materials, assignments, and scoring.
- **In-class demos:** Each module includes brief instructor-led demos, recorded as live SSH-to-server walkthroughs — e.g., a ReAct trace walkthrough (M1), live payoff estimation for a two-agent LLM negotiation (M2/M7), and a live CFR convergence visualization (M5). Demos supplement the graded PAs and are not separately assessed.
- **TAMU LLM API from Week 1:** Every programming assignment uses TAMU's LLM API client ($5/day budget per student), starting with PA1. Students need a TAMU NetID; a setup guide is provided in Week 1.
- **Evaluation discipline:** Introduced in Week 1 and reused in every PA: compute-matched baselines, cost accounting, stochastic variance, model/version pinning, seeds, confidence intervals, and contamination/holdout design.
- **Communication:** All announcements are posted on Canvas. Use Canvas messaging or email for questions.

## 5 Required Materials

- **Textbook:** Shoham & Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations* (freely available online).

- **Supplementary:** Cesa-Bianchi & Lugosi, *Prediction, Learning, and Games* (for regret theory).
- **API access:** TAMU LLM API client, needed from Week 1 — every PA uses it, starting with PA1 (setup guide provided in Week 1; $5/day budget per student).
- **Computing:** Python 3.x, Jupyter, and a standard scientific stack. GPU access is not required.
- **Infrastructure:** OpenSpiel for classical game-solving algorithms; ControlArena for the AI control reproduction (PA5).
- **Paper repository:** All agent-thread papers are provided on the course site, organized by module. See each module's reading list below.

# 6 Grading Policy

| Component | Weight |
|---|---|
| Programming Assignments (5 × 12%) | 60% |
| Course Project | 40% |

**No midterm. No final exam.** Assessment is through paper reproduction and creative research application. Both sections (in-person and asynchronous) share this identical scoring; there is no participation component.

**Grading scale:** A $\geq$ 90% > B $\geq$ 80% > C $\geq$ 70% > D $\geq$ 60% > F.

## 6.1 Late Work Policy

You have **3 slip days** total for programming assignments, usable in 24-hour increments (maximum 2 per assignment). After slip days are used, late submissions are penalized 10% per day, up to 3 days.

# 7 Course Schedule

**Narrative arc:** Define games (M1) → Solve games (M2) → Learn in games (M3) → Handle sequential games (M4) → Solve them at scale (M5) → Make them tractable (M6) → What happens when many agents interact (M7) → Monitor the agents (M8) → Applications and frontiers (M9).

## 7.1 Module 1: Foundations of Strategic Games (Weeks 1–2 — 4 lectures)

**Theory core:** Normal-form games, dominant strategies, Nash equilibrium; mixed strategies, support enumeration; maxmin strategies, minimax theorem; solution concepts: Pareto optimality, social welfare.

**Agent thread:** What is an agent? LLM vs. agentic LLM (the workflow/agent distinction). A ReAct trace as an agent–environment interaction trajectory (not yet a game tree); the induced extensive-form game is formalized in M4 once players, chance, observations, information sets, actions, and utilities are specified. Multi-agent LLM interactions as normal-form games. Demo: a simple

LLM agent loop (observe → reason → act). Evaluation discipline introduced in Week 1 and reused in every PA.

**Reading — Theory:**

- Shoham & Leyton-Brown, *Multiagent Systems*, Ch. 3 (Normal-Form Games).

**Reading — Agent thread:**

- Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models" (ICLR 2023).
- Wang et al., "A Survey on Large Language Model based Autonomous Agents" (2023) — §1–3 as background.
- Xi et al., "The Rise and Potential of Large Language Model Based Agents: A Survey" (2023).
- Anthropic, "Building Effective Agents" (2024) — the workflow vs. agent distinction.
- Sun, Wu, Cheng, and Chu, "Game Theory Meets Large Language Models" (IJCAI 2025) — Week 1 framing paper.

### 7.2 Module 2: Algorithms for Equilibria (Weeks 3–4 — 4 lectures)

**Theory core:** Linear complementarity, Lemke–Howson; correlated equilibrium, coarse correlated equilibrium; linear programming for zero-sum games; complexity of Nash (PPAD-completeness, high level); best response and safe best response.

**Agent thread:** Multi-agent coordination: when two LLM agents interact (debate, negotiation, collaborative coding), what solution concept describes their behavior? Correlated equilibrium via a shared prompt. LLM-based auctions and a mechanism design teaser: do LLM bidders learn to bid truthfully in a second-price auction? (Full treatment in M7.) "What if the opponent isn't rational?" — safe best response as the principled answer.

**Reading — Theory:**

- Shoham & Leyton-Brown, Ch. 4 (Computing Solution Concepts).
- Ganzfried & Sandholm, "Safe Opponent Exploitation" (ACM TEAC 2015).

**Reading — Agent thread:**

- Wellman, "Methods for Empirical Game-Theoretic Analysis" (AAAI 2006).
- Dütting et al., "Mechanism Design for Large Language Models" (WWW 2024).

**PA1 due Week 3** (Empirical Payoff Estimation and Equilibrium Analysis of LLM Agents; see Section 8).

### 7.3 Module 3: Regret Minimization (Week 5 — 2 lectures)

**Theory core:** External, internal, and swap regret; regret matching and regret matching+; no-regret dynamics → coarse correlated equilibrium; Blackwell's approachability (brief).

**Agent thread:** Online tool selection as a bandit/regret problem: an LLM agent choosing which tool to invoke is a repeated decision problem. Adaptive prompt selection via regret minimization. Connection to Nash Learning from Human Feedback (Munos et al., ICML 2024): pairwise preference learning with mirror descent seeks a regularized Nash policy; ordinary RLHF optimizes a learned reward and does not thereby satisfy a no-regret guarantee.

**Reading — Theory:**

- Cesa-Bianchi & Lugosi, *Prediction, Learning, and Games*, Ch. 4 (Regret).
- Farina et al., "Regret Circuits: Composability of Regret Minimizers" (ICML 2019).

**Reading — Agent thread:**

- Schick et al., "Toolformer: Language Models Can Teach Themselves to Use Tools" (NeurIPS 2023).
- Nie et al., "Beyond Numeric Rewards: In-Context Dueling Bandits with LLM Agents" (2024).
- Chen et al., "Which LLM to Play? Convergence-Aware Online Model Selection with Time-Increasing Bandits" (2024).
- Xu et al., "A Component-Based Survey of Interactions between LLMs and Multi-Armed Bandits" (2026).
- Munos et al., "Nash Learning from Human Feedback" (ICML 2024).

**PA2 due Week 7** (Regret Minimization over LLM-Generated Strategies; see Section 8).

### 7.4 Module 4: Extensive-Form Games (Week 6 — 2 lectures)

**Theory core:** Game trees, information sets, strategies; subgame perfect equilibrium, backward induction; imperfect-information games; sequence form, behavioral strategies.

**Agent thread:** An LLM agent's observe → plan → implement → evaluate loop can induce an extensive-form game, but only after defining what is externally observed, what remains private, which nodes belong to chance or another strategic player, and how utility is assigned. Chain-of-thought is a trajectory; Tree of Thoughts is a search tree; neither is automatically a subgame. Planning under uncertainty as imperfect information; recursive agent calls as subgames.

**Reading — Theory:**

- Shoham & Leyton-Brown, Ch. 5 (Extensive-Form Games).

**Reading — Agent thread:**

- Shinn et al., "Reflexion: Language Agents with Verbal Reinforcement Learning" (NeurIPS 2023).
- Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models" (NeurIPS 2022).

- Yao et al., "Tree of Thoughts: Deliberate Problem Solving with Large Language Models" (NeurIPS 2023).
- Zhou et al., "Language Agent Tree Search" (ICML 2024).

**Project Proposal due** at the end of Week 6.

### 7.5 Module 5: CFR and Self-Play (Week 7 — 2 lectures)

**Theory core:** The CFR algorithm and counterfactual values; CFR+ and discounted variants; Monte Carlo CFR (outcome sampling, external sampling); convergence guarantees.

**Agent thread:** Self-play for agent improvement. CFR uses counterfactual reach-weighted local regret updates in a specified extensive-form game. Constitutional AI is supervised self-revision plus RLAIF — not a two-player CFR process. Debate is structurally closer (players, turns, judge), but its training protocols still do not generally implement CFR. We compare what CFR requires (information sets, terminal utility, counterfactual deviations, repeated updates) with what debate and Constitutional AI provide. Monte Carlo CFR as a template for sample-efficient agent evaluation.

**Reading — Theory:**

- Zinkevich et al., "Regret Minimization in Games with Incomplete Information" (NeurIPS 2007).
- Lanctot et al., "Monte Carlo Sampling for Regret Minimization in Extensive Games" (NeurIPS 2009).

**Reading — Agent thread:**

- Irving et al., "AI Safety via Debate" (2018).
- Bai et al., "Constitutional AI: Harmlessness from AI Feedback" (2022).
- Arnesen et al., "Training Language Models to Win Debates with Self-Play Improves Judge Accuracy" (2024).
- Chen et al., "Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models (SPIN)" (ICML 2024).

**PA3 due Week 10** (CFR on Kuhn Poker vs. LLM Self-Play).

### 7.6 Module 6: Abstraction and Agent Action Spaces (Week 8 — 2 lectures)

**Theory core:** Game abstraction: information abstraction, action abstraction; lossy abstraction and solution quality bounds; progressive growing of abstraction.

**Agent thread:** Intent vocabularies as action abstraction: collapsing a huge action space (all possible LLM outputs) into a small intent vocabulary (OBSERVE, PLAN, IMPLEMENT, EVALUATE, TERMINATE) is game abstraction; a non-LLM controller selects intents while the LLM executor fills in details. An intent vocabulary is an abstraction design choice until a mapping and error/loss guarantee are proved — classical bounds do not transfer automatically to arbitrary language actions. The $5/day API budget forces abstraction; finer intents give more control but cost more.

**Reading — Theory:**

- Sandholm, "Abstraction for Solving Large Incomplete-Information Games" (AAAI 2015).

**Reading — Agent thread:**

- Kuhnle & Nath, "LM-AA: Language Model Action Abstraction" (2024/2025).
- Yang et al., "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering" (2024).
- Qin et al., "ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs" (ICLR 2024).
- Schick et al., "Toolformer" (NeurIPS 2023) — also relevant here.

### 7.7 Module 7: Multi-Agent LLM Systems (Week 9 — 2 lectures)

**Theory core:** The empirical game-theoretic analysis (EGTA) pipeline for stochastic LLM policies: strategy specification, simulation design, payoff estimation with confidence intervals, response graphs, approximate equilibrium, and iterative policy discovery. Communication mechanisms: private-value negotiation and debate as protocols. Coordination topologies and failure modes: star, chain, tree, and graph structures.

**Agent thread:** LLM debate, negotiation, and collaborative problem-solving: when two LLM agents debate a factual question over multiple rounds, does the answer improve? (Du et al. 2023: yes.) LLM negotiation with private valuations — does the mechanism elicit truthful reporting? Populations of agents: EGTA analyzes the strategy profiles that emerge from tournaments of LLM agents.

**Reading — Theory:**

- Wellman, "Methods for Empirical Game-Theoretic Analysis" (AAAI 2006) — full treatment here.
- Wellman, Tuyls, and Greenwald, "Empirical Game-Theoretic Analysis: A Survey" (JAIR 2025).

**Reading — Agent thread:**

- Du et al., "Improving Factuality and Reasoning in Language Models through Multiagent Debate" (2023) — the PA4 reproduction target.
- Bianchi et al., "How Well Can LLMs Negotiate? NegotiationArena Platform and Analysis" (2024).
- Mao et al., "Alympics: LLM Agents Meet Game Theory" (2023).
- Hua et al., "Game-Theoretic LLM: Agent Workflow for Negotiation Games" (2024).
- Zhu et al., "MultiAgentBench: Benchmarking Multi-Agent Coordination" (ACL 2025).
- Cemri et al., "Why Do Multi-Agent LLM Systems Fail?" (NeurIPS 2025).

**PA4 due Week 12** (Reproduce Multiagent Debate).

### 7.8 Module 8: Monitoring Autonomous Agents (Week 10 — 2 lectures)

**Theory core:** Submodular optimization: the greedy algorithm and the $(1 - 1/e)$ guarantee for monotone submodular maximization; adaptive submodularity (Golovin & Krause); adversarial submodular games (the Interventionist Greedy Game, IGG); budget-limited intervention; monotone vs. non-monotone objectives.

**Agent thread:** Three monitoring regimes. Regime 1 — reliability monitoring: the agent is fallible but not strategic; interventions cover failures (the safety-coverage function must be defined, with monotonicity and diminishing returns justified, before invoking the greedy guarantee). Regime 2 — adaptive monitoring: inspections reveal information; adaptive submodularity is the natural extension. Regime 3 — AI control (Greenblatt et al.): an untrusted agent may intentionally evade the protocol; designer and red team are strategic players. Why looping happens: an agent repeating the same fix has zero marginal value under the submodular objective. The off-limits model: pruned actions stay pruned. Poison pill construction: non-monotone objectives are quadratically harder.

**Reading — Theory:**

- Krause & Golovin, "Submodular Function Maximization" (*Tractability*, Ch. 3, Cambridge UP, 2014).
- Nemhauser, Wolsey & Fisher, "An Analysis of Approximations for Maximizing Submodular Set Functions—I" (*Mathematical Programming*, 1978).
- Kuhnle, "Interventionist Greedy Game (IGG)" (draft provided in class, 2026).

**Reading — Agent thread:**

- Greenblatt et al., "AI Control: Improving Safety Despite Intentional Subversion" (ICML 2024) — the PA5 reproduction target.
- Chan et al., "Visibility into AI Agents" (FAccT 2024).
- Perez et al., "Red Teaming Language Models with Language Models" (EMNLP 2022).
- Zhan et al., "InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated LLM Agents" (ACL 2024).
- Golovin and Krause, "Adaptive Submodularity" (JAIR 2011).
- Griffin et al., "Games for AI Control" (2024).
- UK AISI/Redwood, ControlArena (software library; PA5 starting point).
- Baker et al., "Monitoring Reasoning Models for Misbehavior" (2025).

**PA5 due Week 14** (Reproduce AI Control Monitoring).

### 7.9 Module 9: Applications, Case Studies, and Frontiers (Weeks 11–12 — 2 lectures)

**Topics** (2–3 per lecture): Poker AI: Libratus, Pluribus, ReBeL (compressed); Cicero/Diplomacy as the language-plus-strategy capstone; AlphaGo, AlphaStar, and self-play at scale; modern agent benchmarks: SWE-bench, WebArena, GAIA; evaluation methodology: cost, variance, contamination, reproducibility.

**Reading — Poker and strategy games:**

- Brown & Sandholm, "Superhuman AI for Multiplayer Poker" (*Science*, 2019).
- FAIR et al., "Cicero: Towards Autonomous AI for Strategy Games" (*Science*, 2022).

**Reading — Agent benchmarks:**

- Jimenez et al., "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" (ICLR 2024).
- Zhou et al., "WebArena: A Realistic Web Environment for Building Autonomous Agents" (ICLR 2024).
- Mialon et al., "GAIA: A Benchmark for General AI Assistants" (ICLR 2024).
- Kapoor et al., "AI Agents That Matter" (TMLR 2025).

**Progress Report due** at the end of Week 11.

### 7.10 Weeks 13–14: PA and Project Crunch

No new lectures. These two weeks are reserved for PA5 completion and course project work: instructor office hours, optional work sessions, and project consultations.

**PA5 due Week 14. Project Presentations in Week 14.**

### 7.11 Week 15: Course Wrap-Up

**Final Project Reports due.** No final exam. The course wraps with project submission and optional feedback sessions.

## 8 Programming Assignments

All PAs are **paper reproduction + analysis** assignments. The implementation is the ticket to entry; the analysis is the assessment. **All five PAs use TAMU's LLM API client** ($5/day budget per student). Students will need a TAMU NetID; API setup instructions are provided in Week 1.

- **PA1 — Empirical Payoff Estimation and Equilibrium Analysis of LLM Agents (due Week 3).** Using the TAMU API, run pairs of LLM agents in simple normal-form settings (prisoner's dilemma, a coordination game, and one negotiation-style game of your choice). Estimate the empirical payoff matrix from repeated play, applying the Week 1 evaluation discipline: seeds, model pinning, variance, and confidence intervals. Then, using provided

implementations of support enumeration and Lemke–Howson, compute Nash equilibria of the estimated games and verify the equilibrium conditions by hand on at least two instances. Analyze: do the LLM agents play equilibrium strategies? How sensitive are your equilibrium conclusions to payoff estimation error? Write a 2-page analysis.

- **PA2 — Regret Minimization over LLM-Generated Strategies (due Week 7).** Implement regret matching (RM) and RM+ and reproduce convergence plots on standard game instances (RPS, asymmetric $3 \times 3$, Shapley's game), in the style of Cesa-Bianchi & Lugosi. Then, using the TAMU API, construct a set of LLM-generated strategies (distinct prompted policies for a repeated matrix game) and run your no-regret algorithms with these strategies as the "experts"/actions, estimating payoffs from LLM policy evaluations. Compare empirical regret against the theoretical bounds and against the classical instances: when does RM+ converge faster than vanilla RM? What is the empirical convergence rate to CCE? Write a 2-page report.

- **PA3 — CFR on Kuhn Poker vs. LLM Self-Play (due Week 10).** Implement vanilla CFR and reproduce exploitability curves on Kuhn poker (Zinkevich et al. 2007). Compare CFR vs. CFR+ vs. MCCFR (outcome sampling). Identify a regime where sampling hurts convergence. Debug a provided MCCFR variant with planted bugs. Then, using the TAMU API, have an LLM play Kuhn poker via self-play (reusing the payoff-estimation harness from PA1) and measure the exploitability of the resulting empirical strategy against your CFR baseline. Write up: convergence analysis, sampling tradeoffs, how far LLM self-play falls from the CFR equilibrium, and the connection to the self-play agent thread from M5.

- **PA4 — Reproduce Multiagent Debate (due Week 12).** Reproduce Du et al., "Improving Factuality and Reasoning in Language Models through Multiagent Debate" (2023). Using the TAMU API, implement the multi-round debate protocol with 2–3 LLM agents on a factual QA benchmark (e.g., TriviaQA or a subset of MMLU). Reproduce the accuracy improvement over a single-agent baseline across 1, 2, and 3 debate rounds. Analyze: can the observed debate protocol be modeled and tested as approaching a CCE under an explicitly defined empirical game (with defined utilities and deviation class)? What happens with 3 agents vs. 2? Does adding a "devil's advocate" role change the dynamics? Mechanism design extension: propose and test one modification to the debate mechanism (e.g., a scoring rule or judge-payment scheme intended to reward truthful argumentation) and analyze how it shifts the equilibrium of the empirical game. Write a 3-page report connecting findings to equilibrium concepts from M2–M3.

- **PA5 — Reproduce AI Control Monitoring (due Week 14).** Reproduce key results from Greenblatt et al., "AI Control: Improving Safety Despite Intentional Subversion" (ICML 2024). Using the TAMU API, implement a "trusted monitor" protocol: a capable LLM performs coding tasks while a monitor LLM flags suspicious actions. Measure the safety–usefulness Pareto frontier by varying the suspicion threshold. Write up: how does monitoring budget (number of actions reviewed) affect the safety–usefulness tradeoff? Connect to the submodular intervention framework from Module 8: define the set of inspected outputs and the safety coverage as a function of that set — does adding one more inspection have diminishing marginal benefit, and under what assumptions? This PA builds directly on PA1–PA4: reuse the payoff-estimation harness (PA1), the empirical evaluation discipline (PA1–PA2), and the multi-agent protocol machinery (PA4). Consider using ControlArena (UK AISI/Redwood) as the reproduction harness rather than building from scratch.

# 9 Course Project

Apply a game-theoretic concept from this course to a multi-agent or human–agent interaction involving LLMs. **This is where you get to be creative.**

**Example topics:**

- Build and evaluate a multi-agent debate system using regret minimization.
- Implement a human-on-the-loop monitoring system for an LLM coding agent and analyze its game-theoretic properties.
- Design an auction mechanism for LLM agents and measure how closely they approximate equilibrium play.
- Red-team an LLM agent and formalize the attack as an exploitation strategy.
- Compare CFR self-play to LLM self-play (Constitutional AI) on a structured task.
- Extend the multiagent debate protocol (PA4) with a mechanism design twist.

**Deliverables:**

- Project Proposal — due Week 6
- Progress Report — due Week 11
- Final Presentation — Week 14
- Final Report — Week 15

# 10 Major Dates

| Item | Due | Type |
|---|---|---|
| PA1 (LLM Payoff Estimation + Equilibria) | Week 3 | Programming |
| Project Proposal | Week 6 | Writing |
| PA2 (RM/RM+ over LLM Strategies) | Week 7 | Programming |
| PA3 (CFR on Kuhn Poker) | Week 10 | Programming |
| Progress Report | Week 11 | Writing |
| PA4 (Multiagent Debate) | Week 12 | Programming |
| PA5 (AI Control Monitoring) | Week 14 | Programming |
| Project Presentation | Week 14 | Presentation |
| Project Report | Week 15 | Writing |

# 11 University Policies

## 11.1 Attendance Policy

The university views class attendance and participation as an individual student responsibility. Students are expected to attend class and to complete all assignments. Please refer to Student Rule

7 in its entirety for information about excused absences, including definitions, documentation, and related timelines.

*Note:* The asynchronous section has no synchronous meetings; "attendance" in that context means engaging with the lecture recordings and course materials each week. Attendance does not contribute to the course grade in either section.

### 11.2 Makeup Work Policy

Students will be excused from attending class on the day of a graded activity or when attendance contributes to a student's grade, for the reasons stated in Student Rule 7, or other reason deemed appropriate by the instructor. Please refer to Student Rule 7 in its entirety for information about makeup work, including definitions, and related documentation and timelines.

### 11.3 Academic Integrity Statement and Policy

"An Aggie does not lie, cheat or steal, or tolerate those who do."

Texas A&M University students are responsible for authenticating all work submitted to an instructor. If asked, students must be able to produce proof that the item submitted is indeed the work of that student. Students must keep appropriate records at all times. The inability to authenticate one's work, should the instructor request it, may be sufficient grounds to initiate an academic misconduct case (Student Rule 20.1.2.3).

You may use LLM tools (ChatGPT, Claude, Copilot, etc.) for programming assignments. However, you must understand and be able to explain all code you submit. The analysis and write-up portions of each PA must be your own original work. You can learn more about the Aggie Honor System Office Rules and Procedures, academic integrity, and your rights and responsibilities at aggiehonor.tamu.edu.

### 11.4 Americans with Disabilities Act (ADA) Policy

Texas A&M University is committed to providing equitable access to learning opportunities for all students. Students who experience barriers to their education due to a disability or who think they may have a disability should contact Disability Resources as early as possible. For College Station and most system locations, Disability Resources can be reached at (979) 845-1637 or disability@tamu.edu. Disabilities may include, but are not limited to, attentional, learning, mental health, sensory, physical, or chronic health conditions. All students are encouraged to discuss their disability-related needs with Disability Resources and their instructors as soon as possible.

### 11.5 Notice of Nondiscrimination

Texas A&M University is committed to providing safe and non-discriminatory learning, living, and work environments for all members of the University community. The University provides equal opportunity to all employees, students, applicants for employment or admission, and the public, regardless of race, color, sex (including pregnancy and related conditions), religion, national origin, age, disability, genetic information, or veteran status.

For more information on civil rights and Title IX, see civilrights.tamu.edu.

### 11.6 Pregnancy Accommodations

Texas A&M provides reasonable accommodations to students due to pregnancy and related conditions, such as childbirth, recovery, and lactation. Students should contact the University's Pregnancy Coordinator as soon as they become aware of the need for accommodation.

### 11.7 Mental Health and Wellness

Texas A&M University recognizes that mental health and wellness are critical factors influencing academic success and overall wellbeing. Students are encouraged to utilize University Health Services and other campus resources as needed. If you or someone you know is experiencing mental health concerns, please contact Counseling & Psychological Services (CAPS) at (979) 845-4427 or visit caps.tamu.edu.

In an emergency or life-threatening situation, call 911 or go to the nearest emergency room. The 988 Suicide & Crisis Lifeline is available 24/7 by calling or texting 988 or via 988lifeline.org.

### 11.8 Family Educational Rights and Privacy Act (FERPA)

FERPA is a federal law designed to protect the privacy of educational records. For more information about student rights under FERPA at Texas A&M, see the university's FERPA resources.