Connecticut AI Alliance (CAIA)
AI Computing Access Program

Sample Applications: AI Course Delivery Support Track

Award Period: Summer 2026 – Fall 2026

Note to Applicants: *The following two sample applications are provided to illustrate the level of detail and specificity expected in a competitive submission to the AI Course Delivery Support track. All names, institutions, and contact information are fictional. These samples represent common application scenarios and are not intended to be copied verbatim. Reviewers value clear alignment between computing needs and course learning objectives, realistic per-student resource estimates, and plans for sharing or reusing developed materials across the CAIA community.*

Sample Application 1: Graduate NLP Course with LLM Lab Components

Scenario: *A data science lecturer at a smaller CAIA member institution is teaching a graduate Natural Language Processing course for the first time in Fall 2026. The course integrates LLM API access throughout its assignments and a final project, and needs GPU resources for a fine-tuning module. The institution has no GPU infrastructure. This sample illustrates how to present a new course with detailed per-assignment resource estimates.*

Section 1: Instructor Information

1. Full Name

Dr. Patricia Mwangi

2. Email Address

pmwangi@havendale.edu

3. CAIA Member Institution

Havendale College

4. Department / Program

Department of Mathematics and Data Science

5. Title / Rank

Lecturer

Section 2: Course Information

6. Course Title

Natural Language Processing with Large Language Models

7. Course Number

DSCI 5250

8. Course Level

- ■ Undergraduate (100–400 level)
- ■ Graduate (500–600+ level)
- ■ Combined undergraduate/graduate section

9. When will this course be offered?

- ■ Summer 2026
- ■ Fall 2026
- ■ Both Summer + Fall 2026

10. Expected Enrollment

18 students

11. Number of Sections

1

12. Is this a new course or an existing course?

- ■ New course – first time being offered
- ■ Existing course – being taught with new AI computing component
- ■ Existing course – computing resources previously available through other means

13. Course Description

DSCI 5250 is a new graduate course that introduces students to the theory and practice of natural language processing with a focus on modern large language models. The course is designed for MS students in Data Science, Computer Science, and related programs who have completed at least one course in machine learning or statistical modeling. The curriculum is organized into four units: (1) Foundations of text representation and classical NLP (weeks 1–3: tokenization, embeddings, sequence models); (2) Transformer architectures and pre-trained language models (weeks 4–6: attention mechanisms, BERT, GPT, encoder vs. decoder architectures); (3) Applied LLM techniques (weeks 7–10: prompt engineering, retrieval-augmented generation, fine-tuning with LoRA, evaluation and benchmarking); and (4) Responsible NLP and emerging topics (weeks 11–13: bias in language models, hallucination detection, multilingual NLP, agents). Each unit includes a hands-on lab and a graded assignment that requires students to interact with LLMs via API or train/fine-tune models on GPU hardware. The course culminates in a three-week final project in which student teams build and evaluate an NLP application of their choice.

14. Course Syllabus Upload

Uploaded (draft syllabus, PDF).

Section 3: Computing Resource Integration

15. How will cloud computing resources be used in this course?

- ■ Hands-on assignments / homework (students train or fine-tune models)
- ■ In-class labs or live coding demonstrations
- ■ Course projects (multi-week student projects requiring compute)
- ■ LLM-powered tools integrated into coursework (e.g., RAG, chatbot, code assistant)
- ■ Student experimentation with pre-trained models (inference only)
- ■ Automated grading or assessment using AI tools
- ■ Other

16. Describe the specific assignments, labs, or activities that require computing resources.

The course includes four graded assignments and a final project, each with specific computing needs:

Assignment 1 (Week 3): Word Embeddings and Text Classification. Students train Word2Vec and GloVe embeddings on a provided corpus (~500K sentences) and build a text classifier using a simple feedforward network. Framework: PyTorch. Compute: CPU-only (Colab free tier is sufficient for this assignment; included here for completeness).

Assignment 2 (Week 6): Transformer Fine-Tuning. Students fine-tune a pre-trained DistilBERT model on a sentiment analysis task (SST-2 dataset, ~67K samples). Framework: Hugging Face Transformers + PyTorch. Compute: T4 GPU required; estimated 20–30 minutes of GPU time per student for training plus experimentation.

Assignment 3 (Week 9): Prompt Engineering and RAG Pipeline. Students build a retrieval-augmented generation pipeline using LangChain, a vector database (ChromaDB), and the Gemini Pro API. They design and evaluate prompts for a domain-specific Q&A task using a provided document corpus. Framework: LangChain, ChromaDB, Vertex AI SDK. Compute: LLM API access; estimated 150K–200K input tokens and 50K–75K output tokens per student for development and evaluation.

Assignment 4 (Week 11): LoRA Fine-Tuning of an LLM. Students fine-tune a 3B-parameter open-weight model (e.g., Phi-3-mini or Gemma-2B) using LoRA adapters on a task of their choice from a provided menu (summarization, instruction following, or classification). Framework: Hugging Face PEFT + PyTorch. Compute: T4 GPU; estimated 60–90 minutes of GPU time per student.

Final Project (Weeks 11–14). Teams of 2–3 students build an NLP application that combines at least two techniques from the course (e.g., fine-tuned model + RAG pipeline, or custom embedding + LLM evaluation). Compute: T4 GPU + LLM API; estimated 2–3 hours of GPU time per team and 300K–500K API tokens per team.

In-Class Labs (6 sessions). Live coding demonstrations where the instructor walks through model training, API integration, and evaluation workflows. Students follow along in their own Colab notebooks. Compute: T4 GPU for ~30 minutes per session per student; API access for 2 sessions (~50K tokens per student per lab session).

17. What type(s) of computing resources are you requesting?

- ■ GPU Computing – Google Colab Pro/Pro+ (individual student accounts)
- ■ GPU Computing – JupyterHub (shared managed environment)
- ■ LLM API Access – Google Vertex AI

18. If requesting GPU computing, what level of GPU access is needed per student?

- ■ Basic (T4, 16 GB VRAM) – sufficient for most coursework
- ■ Standard (A100 40 GB) – needed for larger model training in advanced courses
- ■ Flexible / instructor will manage allocation

19a. Which models will students use?

Gemini Pro (primary, for RAG pipeline assignment and final projects); PaLM 2 (for text classification comparisons in Assignment 3).

19b. Estimated API usage per student per semester

Approximately 400K–600K input tokens and 150K–200K output tokens per student over the full semester. This includes all assignments, labs, and the final project. Breakdown: Assignment 3 (~200K input), final project (~250K input), in-class labs (~100K input), informal experimentation buffer (~50K input).

19c. How will API access be distributed to students?

- ■ Shared project-level API key managed by instructor
- ■ Individual student API keys
- ■ Through a course application/interface (e.g., custom chatbot, Jupyter notebook)
- ■ Other

API access will be distributed through pre-configured Colab notebooks with the Vertex AI SDK. The instructor will manage a single project-level service account; each student notebook authenticates via a shared credential scoped to the course project. Per-student usage will be tracked using custom logging wrappers built into the notebook templates. Budget alerts at 80% and 95% of per-student allocations will notify the instructor.

20. Estimated total cloud credit budget requested

$3,200 ($1,800 for Colab Pro+ subscriptions for 18 students for ~14 weeks; $1,400 for Vertex AI API credits across all assignments, labs, and final projects). Formula: 18 students x $100/student Colab Pro+ allocation = $1,800; 18 students x ~$75/student API usage estimate = $1,350, rounded to $1,400 with buffer.

Section 4: Pedagogical Value & Student Outcomes

21. How will access to these computing resources improve student learning outcomes?

Havendale College does not currently have any GPU-equipped computing resources available for instruction. Students in our existing data science courses have been limited to working with small datasets on CPU-only environments, which has made it impossible to include meaningful hands-on deep learning or LLM interaction in the curriculum. Without GPU access, the fine-tuning assignments (Assignments 2 and 4) would be impractical: fine-tuning even a small DistilBERT model on CPU takes over 6 hours, far beyond what students can manage in a weekly assignment cycle. Without LLM API access, the RAG pipeline assignment and any project involving LLM integration would need to rely on small, locally hosted models that do not reflect the capabilities or behavior of the models students will encounter professionally. Cloud access transforms this course from a primarily theoretical survey into a hands-on, skills-building experience. Students will graduate having trained real models on real datasets, built functional LLM-powered applications, and managed cloud computing budgets. These are precisely the competencies our industry advisory board has identified as gaps in our current graduates' preparation.

22. What AI/ML competencies will students develop?

- ■ Model training and evaluation (supervised, unsupervised, reinforcement learning)
- ■ Deep learning with neural networks (CNNs, RNNs, Transformers)
- ■ LLM prompt engineering and application development
- ■ Data preprocessing and feature engineering at scale
- ■ Model deployment and inference optimization
- ■ Experiment tracking, hyperparameter tuning, and MLOps basics
- ■ Responsible AI practices (bias detection, fairness evaluation, interpretability)
- ■ Working with cloud computing infrastructure
- ■ Other

23. Will course materials be shared or reusable?

- ■ Yes – assignments/labs will be shared via CAIA's resource repository
- ■ Yes – Jupyter notebooks or Colab notebooks will be published (e.g., GitHub)
- ■ Yes – course modules can serve as templates for other CAIA institutions
- ■ Possibly – willing to share if materials are refined
- ■ No – materials are institution-specific

All assignment notebooks and lab materials will be published on a public GitHub repository under a CC BY-SA 4.0 license at the end of the semester. The instructor is also willing to present a workshop on course design and materials at a future CAIA meeting to support adoption at other member institutions.

24. How will you assess the impact of computing access on student learning?

The instructor will administer a pre-course and post-course self-assessment survey measuring students' confidence with GPU computing, LLM API usage, and deep learning workflows. Assignment scores on compute-dependent assignments will be compared to analogous (CPU-only) assignments in the department's existing ML course. End-of-semester course evaluations will include targeted questions about the computing resources. The instructor will summarize findings in the required CAIA end-of-semester report.

Section 5: Technical Readiness & Course Logistics

25. What is the current computing setup for this course?

- ■ No computing resources currently available for AI/ML workloads
- ■ Limited local/institutional resources
- ■ Some cloud access through another provider
- ■ Students use personal machines
- ■ Other

26. Do you or your students have prior experience with the requested cloud platform(s)?

- ■ Yes – instructor is experienced; students will be onboarded
- ■ Partial – instructor has some experience; will need setup guidance
- ■ No – both instructor and students are new to the platform

The instructor has used Google Colab extensively in personal research and has taught with Colab free-tier notebooks in prior courses. Students will be onboarded during Week 1 with a dedicated setup lab that covers Colab Pro features, GPU runtime management, and cost awareness.

27. Will you need TA support for managing computing resources?

- ■ Yes – I have a TA who will help manage student access and troubleshoot
- ■ No – I will manage resources directly
- ■ I would benefit from TA support but do not currently have one

Section 6: Responsible Use & Data Practices

28. Will students work with any of the following?

- ■ Real-world datasets containing PII or sensitive information
- ■ Web-scraped or user-generated content
- ■ Synthetic media generation (text, image, audio, video)
- ■ Health, financial, or legal data
- ■ None of the above

Students will generate synthetic text using LLM APIs as part of the RAG and prompt engineering assignments. Final projects may also involve text generation.

29. How will you ensure responsible use of AI computing resources by students?

All students will sign a computing resource acceptable use agreement during Week 1, which covers academic integrity expectations (no using LLM outputs in other courses without disclosure), prohibited uses (generating harmful or deceptive content), and resource conservation practices (disconnecting GPU runtimes when not in use). The course syllabus includes a dedicated unit on Responsible NLP (Unit 4, weeks 11–13) covering bias in language models, hallucination risks, and ethical deployment considerations. Each assignment rubric includes a "Limitations and Ethics" section worth 10% of the assignment grade, requiring students to reflect on the risks and limitations of their work. API usage is monitored weekly by the TA using the logging wrappers in the notebook templates.

Section 7: Prior Support

30. Have you previously received computing resources through CAIA?

- ■ Yes
- ■ No

Section 8: Applicant Certification

31. Electronic Signature

Patricia Mwangi

32. Date

May 10, 2026

Sample Application 2: Undergraduate Intro to AI at a Community College

Scenario: *An adjunct professor at a community college is teaching an introductory AI course to a large undergraduate section. Students have limited technical backgrounds and no access to GPU hardware. The instructor requests primarily LLM API access so students can interact with AI models through structured notebook exercises, plus modest GPU access for a supervised learning module. This sample illustrates a lower-budget, access-focused application from a two-year institution.*

Section 1: Instructor Information

1. Full Name

Prof. David Nguyen

2. Email Address

dnguyen@valleytech.edu

3. CAIA Member Institution

Valley Technical Community College

4. Department / Program

Department of Computer Information Systems

5. Title / Rank

Adjunct Professor

Section 2: Course Information

6. Course Title

Introduction to Artificial Intelligence

7. Course Number

CIS 2180

8. Course Level

- ■ Undergraduate (100–400 level)
- ■ Graduate (500–600+ level)
- ■ Combined undergraduate/graduate section

9. When will this course be offered?

- ■ Summer 2026
- ■ Fall 2026
- ■ Both Summer + Fall 2026

10. Expected Enrollment

35 students

11. Number of Sections

1

12. Is this a new course or an existing course?

- ■ New course – first time being offered
- ■ Existing course – being taught with new AI computing component
- ■ Existing course – computing resources previously available through other means

CIS 2180 has been taught for three semesters using free-tier Colab and publicly available demos. The new computing component will replace limited free-tier access with Colab Pro for the image classification module and add LLM API integration for three new assignments that were previously impossible to include.

13. Course Description

CIS 2180 is an introductory AI course for undergraduate students in Computer Information Systems, Computer Science, and other STEM-adjacent programs. It is designed to be accessible to students who have completed one semester of Python programming (CIS 1110 or equivalent). The course provides a broad survey of AI concepts, techniques, and applications organized around five modules: (1) What Is AI? (history, definitions, societal impact); (2) Search and Problem Solving (state spaces, heuristics, A* search); (3) Machine Learning Fundamentals (supervised learning, classification, regression, evaluation metrics); (4) Neural Networks and Deep Learning (perceptrons, backpropagation, CNNs for image classification); and (5) Language Models and Generative AI (how LLMs work, prompt design, RAG concepts, limitations and risks). The course emphasizes hands-on exploration over mathematical rigor; students interact with pre-built models, modify starter code in guided notebooks, and complete structured exercises rather than building models from scratch. The computing resources requested here support Modules 4 and 5, which require GPU access for CNN training and LLM API access for the language model exercises.

14. Course Syllabus Upload

Uploaded (current syllabus with proposed revisions highlighted, PDF).

Section 3: Computing Resource Integration

15. How will cloud computing resources be used in this course?

- ■ Hands-on assignments / homework (students train or fine-tune models)
- ■ In-class labs or live coding demonstrations
- ■ Course projects (multi-week student projects requiring compute)
- ■ LLM-powered tools integrated into coursework (e.g., RAG, chatbot, code assistant)
- ■ Student experimentation with pre-trained models (inference only)
- ■ Automated grading or assessment using AI tools
- ■ Other

16. Describe the specific assignments, labs, or activities that require computing resources.

Computing resources are needed for four specific course components:

Lab 4 (Week 8): Image Classification with a CNN. Students work through a guided Colab notebook to train a small CNN (3 convolutional layers, ~200K parameters) on the CIFAR-10 dataset. The notebook provides all code; students modify hyperparameters (learning rate, batch size, number of epochs), observe how changes affect accuracy, and answer guided reflection questions. Framework: PyTorch (pre-installed in Colab). Compute: T4 GPU; each student runs 3–5 training experiments of approximately 5 minutes each, for a total of ~15–25 minutes of GPU time per student.

Assignment 3 (Week 10): Transfer Learning Exercise. Students load a pre-trained ResNet-18 model and fine-tune the final classification layer on a small custom dataset (flowers dataset, ~3,600 images). They compare transfer learning performance to training from scratch (using their Lab 4 experience as a reference point). Compute: T4 GPU; estimated 10–15 minutes of GPU time per student.

Lab 6 (Week 12): Exploring LLM Capabilities and Limitations. Students interact with Gemini Pro through a structured Colab notebook to complete exercises on prompt design, zero-shot vs. few-shot classification, and hallucination identification. The notebook provides scaffolded prompts that students modify and extend. Compute: LLM API; estimated 40K–60K input tokens and 15K–25K output tokens per student.

Assignment 4 (Week 13): Build a Simple Q&A Chatbot. Students use a provided template notebook to build a minimal RAG-style Q&A system over a small document collection (5–10 provided PDF articles on a topic of their choice from a curated list). The assignment emphasizes prompt engineering, retrieval quality, and answer evaluation rather than infrastructure. Framework: LangChain (simplified wrapper provided), Vertex AI SDK. Compute: LLM API; estimated 80K–120K input tokens and 30K–50K output tokens per student (including iterative testing).

17. What type(s) of computing resources are you requesting?

- ■ GPU Computing – Google Colab Pro/Pro+ (individual student accounts)
- ■ GPU Computing – JupyterHub (shared managed environment)
- ■ LLM API Access – Google Vertex AI

18. If requesting GPU computing, what level of GPU access is needed per student?

- ■ Basic (T4, 16 GB VRAM) – sufficient for most coursework
- ■ Standard (A100 40 GB)
- ■ Flexible / instructor will manage allocation

19a. Which models will students use?

Gemini Pro (all LLM exercises and the Q&A chatbot assignment).

19b. Estimated API usage per student per semester

Approximately 120K–180K input tokens and 45K–75K output tokens per student over the semester. This is a conservative estimate; usage is concentrated in weeks 12–14. Breakdown: Lab 6 (~50K input), Assignment 4 (~100K input), informal experimentation buffer (~30K input).

19c. How will API access be distributed to students?

- ■ Shared project-level API key managed by instructor
- ■ Individual student API keys
- ■ Through a course application/interface (e.g., custom chatbot, Jupyter notebook)
- ■ Other

All API access is mediated through pre-configured Colab notebooks with a shared service account credential. Students never see or handle API keys directly. A simple logging function in each notebook records per-student usage to a shared Google Sheet that the instructor monitors weekly.

20. Estimated total cloud credit budget requested

$1,900 ($1,050 for Colab Pro for 35 students for the 6-week period when GPU access is needed [weeks 8–14, approximately $30/student]; $850 for Vertex AI API credits across 35 students). Formula: 35 students x $30 Colab = $1,050; 35 students x ~$24/student API estimate = $840, rounded to $850.

Section 4: Pedagogical Value & Student Outcomes

21. How will access to these computing resources improve student learning outcomes?

Valley Technical Community College serves a diverse student body, many of whom are first-generation college students, career changers, or working adults pursuing associate or bachelor's degrees. Most students do not own computers capable of running GPU-accelerated workloads, and the college's computer labs are equipped with standard office PCs without dedicated GPUs. In past semesters, the deep learning module (Module 4) has been taught almost entirely through lecture slides and pre-recorded demonstrations because students could not run training experiments on their own hardware. As a result, student survey feedback consistently describes Module 4 as the least engaging part of the course. With Colab Pro access, every student will be able to train a CNN and observe how architectural and hyperparameter choices affect model performance in real time. This transforms a passive unit into an active, exploratory one. Similarly, the LLM exercises in Module 5 have been limited to interacting with free chatbot interfaces, which do not allow students to examine how prompt structure, temperature, and retrieval context affect model outputs programmatically. API access gives students a structured, code-based interface to these models, building skills that transfer directly to entry-level data science and software development roles.

22. What AI/ML competencies will students develop?

- ■ Model training and evaluation (supervised, unsupervised, reinforcement learning)
- ■ Deep learning with neural networks (CNNs, RNNs, Transformers)
- ■ LLM prompt engineering and application development
- ■ Data preprocessing and feature engineering at scale
- ■ Model deployment and inference optimization
- ■ Experiment tracking, hyperparameter tuning, and MLOps basics
- ■ Responsible AI practices (bias detection, fairness evaluation, interpretability)
- ■ Working with cloud computing infrastructure
- ■ Other

23. Will course materials be shared or reusable?

- ■ Yes – assignments/labs will be shared via CAIA's resource repository
- ■ Yes – Jupyter notebooks or Colab notebooks will be published (e.g., GitHub)
- ■ Yes – course modules can serve as templates for other CAIA institutions
- ■ Possibly
- ■ No

This course was specifically designed to be adoptable by other two-year and four-year institutions in the CAIA network. All notebooks are scaffolded with extensive comments and instructor notes. The instructor is committed to contributing the full set of notebooks and assignment rubrics to CAIA's resource repository and is willing to co-lead an adoption workshop for interested faculty at other community colleges.

24. How will you assess the impact of computing access on student learning?

A short pre/post confidence survey (adapted from the CS self-efficacy scale) will measure changes in students' comfort with AI tools, GPU computing, and LLM interaction. Assignment scores on the compute-dependent assignments will be compared to performance on the non-compute assignments as a within-course benchmark. End-of-semester evaluations will include three targeted questions on the computing resources. Results will be summarized in the CAIA end-of-semester report.

Section 5: Technical Readiness & Course Logistics

25. What is the current computing setup for this course?

- ■ No computing resources currently available for AI/ML workloads
- ■ Limited local/institutional resources
- ■ Some cloud access through another provider
- ■ Students use personal machines (insufficient for course objectives)
- ■ Other

Students currently use personal laptops or college lab PCs with Colab free tier. Free-tier GPU access is unreliable (frequent disconnections, session limits) and unavailable during peak usage periods, making it unsuitable for graded assignments with deadlines.

26. Do you or your students have prior experience with the requested cloud platform(s)?

- ■ Yes – instructor is experienced; students will be onboarded
- ■ Partial

- ■ No

The instructor has used Colab Pro in personal projects and in a previous adjunct position. Students are introduced to Colab (free tier) in Week 1 of the course; the transition to Pro features will be covered in a 20-minute orientation at the start of the GPU module.

27. Will you need TA support for managing computing resources?

- ■ Yes
- ■ No
- ■ I would benefit from TA support but do not currently have one

As an adjunct, the instructor does not have a TA. A peer tutor from the college's Academic Success Center has volunteered to assist with basic Colab troubleshooting during open lab hours. The instructor will handle all credential management and usage monitoring directly.

Section 6: Responsible Use & Data Practices

28. Will students work with any of the following?

- ■ Real-world datasets containing PII or sensitive information
- ■ Web-scraped or user-generated content
- ■ Synthetic media generation (text, image, audio, video)
- ■ Health, financial, or legal data
- ■ None of the above

Students will generate text responses via LLM API calls in Labs 6 and Assignment 4.

29. How will you ensure responsible use of AI computing resources by students?

Students sign a one-page acceptable use agreement during the Week 1 orientation, covering prohibited uses (generating harmful, deceptive, or academically dishonest content), resource conservation (closing GPU runtimes after use), and academic integrity expectations. The course includes a dedicated class session on AI ethics and societal impact (Module 1, Week 3) and a reflective writing assignment asking students to evaluate the risks of a real-world AI deployment. The hallucination identification exercise in Lab 6 is specifically designed to build critical evaluation skills. API usage is logged and reviewed weekly by the instructor to identify any unusual patterns.

Section 7: Prior Support

30. Have you previously received computing resources through CAIA?

- ■ Yes
- ■ No

Section 8: Applicant Certification

31. Electronic Signature

David Nguyen

32. Date

May 18, 2026