+1 (407)-922-9622 ssuarezsolis@calarts.edu

touch to start, a work by a lesson student made on p5.js

THESIS My teaching is a studio-and-lab practice where artists learn to build accountable technological systems in service of expressive work. I treat AI not as a style generator, but as material: something to be instrumented, critiqued, constrained, and performed. Across courses, students develop craft (signal flow, code literacy, audio/visual pipelines), critical fluency (history, politics, ethics of datasets and models), and a documentation habit that makes artistic process legible and repeatable. Critique is dialogic and iterative—grounded in what the work is doing, what the system enables or forecloses, and how the artist authors meaning through choices, constraints, and responsibility.

CONTENTS

+1 (407)-922-9622 ssuarezsolis@calarts.edu

TEACHING PHILOSOPHY In my teaching, students respond best when I either learn alongside them or design conditions in which they learn to teach themselves. In both cases, my responsibility is to help students move repeatedly between competence and experimentation –between what they can do confidently and what they can only reach through curiosity, risk, and iteration. I think of learning as a kind of scaffolding: each new technique, concept, or historical reference becomes a rung that supports the next question. Over time, students build an internal method for making and evaluating work, one they can carry beyond any single class, tool, or aesthetic.

Because composition and media-making are not only technical practices but also practices of meaning, my approach to critique begins with a premise: the student is already an author, not a novice waiting for authorization. Especially at the graduate level, the most urgent questions are rarely "Did you apply the technique correctly?" but "What is this work insisting on?" and "What does it ask of an audience, a space, a community, a history?" I structure critique to surface intention, context, and consequence. One core framework I draw on is the Lerman/Borstel Creative Response Process, which replaces judgment-first critique with dialogic reflection, artist-led questions, and feedback that supports both rigor and agency. This process also scales across disciplines: it works as well for a score-in-progress as it does for an installation, a performance system, a dataset-driven piece, or a hybrid work that doesn't fit inherited categories. In the end, we are making artists, not technicians.

In studio and lesson contexts, I aim to make the classroom a site of practice, reflection, and durable skill-building. Students might analyze scores, recordings, and artistic lineages one week, then translate those observations into constraints for a new study the next. They might prototype quickly, document what happened, and return to revise with clearer intention. I use short, repeatable cycles (proposal → prototype → feedback → revision → reflection) because they mirror professional creative workflows and because they teach students that craft is not separate from thought. To keep the work from drifting into either pure concept or pure technique, I ask for small, consistent "evidence trails": process notes, version histories, listening/viewing journals, and brief reflection memos that name what changed and why.

My technology teaching is digital-forward, but never technology-for-technology's sake. I introduce computation as a material and as a collaborator: a set of constraints, affordances, and ethical problems that shape artistic choices. Students work in creative coding and media environments such as Max/MSP, TouchDesigner, ChucK, SuperCollider, DAWs, and general-purpose languages (e.g., JavaScript/TypeScript), depending on their goals and prior experience. I build courses with multiple onramps: template patches, starter repos, "minimum viable" studies; this way, students at different levels can enter the same assignment without flattening their ambition. As they advance, I emphasize professional habits that make work sustainable: documentation, provenance, maintainability, and reproducibility. Students learn to organize repositories, write clear README files, track versions, and articulate technical decisions in language that collaborators outside their discipline can understand.

When we engage AI tools (language models, generative media systems, or agentic workflows) I frame them as both instruments and objects of critique. Students learn structured prompting as a literacy: how to specify constraints, evaluate outputs, identify failure modes, and iterate responsibly. I sometimes teach a tagged, loop-based prompting method I developed (Viable Prompt Protocol) as one option among several for making interactions with AI systems more predictable and auditable. Just as importantly, we treat AI as a cultural and political force with realworld implications. In practice, that means students account for training data and authorship questions, bias and representational harm, labor and environmental costs, accessibility impacts, and the ethics of deployment. Rather than outsourcing creative decisions to a system, the goal is to sharpen the student's own authorship: to decide when machine assistance clarifies an intention and when it dulls it, and to be able to explain that difference publicly and rigorously.

Across all of this, I work to reduce gatekeeping. I tell students directly: we are conceptually on equal footing, what

differs is not worth, but experience and access to certain forms of knowledge. I teach generously, share resources transparently, and demystify professional practices (from applying to festivals and residencies to preparing portfolios, writing statements, and communicating with collaborators). In advising, I try to anticipate the questions students don't yet know to ask, especially students navigating institutions that were not designed for them. That commitment also shapes course structure: clear expectations, accessible pathways to excellence, flexible modes of participation, and consistent feedback rhythms that support students balancing work, family, and uneven prior preparation.

Finally, I root experimentation in history, craft, and critique. Whether we are studying orchestration, extended techniques, algorithmic processes, or AI-mediated practice, we situate tools within lineages: aesthetic movements, technological histories, and the ethical debates that accompany new media. My aim is for students to graduate with (1) technical fluency; (2) critical vocabulary; (3) an ability to collaborate across disciplines; and (4) a resilient artistic identity: a clear sense of what they are making, why it matters, and how to keep making it under real-world conditions. If my students leave with anything, I want it to be this: the capacity to build their own scaffolds, to ask better questions over time, and to make work that is both formally rigorous and meaningfully alive to the world.

PEDAGOGY My studio pedagogy centers around a weekly cycle that encourages revision, recursion, and iteration, mimicking machine operations cycles; students learn concepts and tools in order to make work, moving through rapid cycles of prototyping, critique, revision, and publication, so technique, aesthetics, and ethics stay fused to practice.

INPUTS

- Curated stimuli that seed making
- Readings: short theory/criticism; methods; ethics/contexts
- Listening / viewing: exemplars + "anti-exemplars" (what to avoid / revise)
- Tool demos: just-in-time, constraint-based (one technique → one artistic choice)
- Artist case studies: process notes, failures, versions, decision logs

THE STUDIO CYCLE (weekly loop)

1. Prompt + constraints (aesthetic aim + technical limit + audience/context)
2. Prototype sprint
3. Critique cycle (structured, dialogic; student-led; evidence-based language)
4. Revision (target 1–2 priorities; iterate versions; test against constraints)
5. Peer review (rubric + reciprocity: feedback must be actionable + transferable)
6. Reflection (why it works / doesn't; what changed; what carries to the next piece)
7. Publish (shareable artifact + documentation; archive for reuse/remix)

OUTPUTS

- Portfolio-ready work + a traceable practice
- Artifact(s): composition / performance / installation / media piece / tool
- Documentation: process log, version trail, screenshots/audio/video, citations
- Artist statement: intention, context, methods, stakes, influences, ethics
- Reproducible workflow: settings, patch/session files, pipeline notes, templates

+1 (407)-922-9622 ssuarezsolis@calarts.edu

SAMPLE ASSIGNMENT PROMPTS

Assignment 1 *Dataset as Score: Building a Reproducible Creative Corpus + Small Model*
Premise: Treat data collection/curation as composition. You will build a small, purpose-built dataset (audio + text + annotations, or any multimodal pairing you can ethically obtain) and use it to drive a creative system. The "piece" is inseparable from the dataset's structure, omissions, and metadata.

Learning objectives

1. Design a dataset intentionally for a creative goal (scope, schema, representativeness, failure modes).
2. Build a reproducible ML pipeline (versioned data, scripted preprocessing, training, evaluation, and artifact logging).
3. Translate model behavior into artistic form through creative coding (mapping, constraints, dramaturgy, iteration).

Constraints

- The dataset must embody an aesthetic argument: it should sound/behave differently if curated differently (and you must prove this).
- Your model must be deliberately small (or narrow): constrained capacity forces craft in data and mapping rather than "bigger model = better output."
- No "generate-and-dump." The system must be compositional: rules, mappings, and timing decisions are authored and documented.

Deliverables (artifact + docs + reflection)

Artifact: 6–10 minute performance, installation, or interactive work powered by your dataset + model (live or rendered).

1. Repo: A public or private repository containing:
    - Data schema + folder structure
    - Preprocessing scripts/notebooks (fully runnable)
    - Training script + config
    - Evaluation script + outputs (plots/logs)
2. Documentation:
    - Data Card (what it is, how gathered, what's excluded, licensing/consent, known biases)
    - Model Card (architecture, training setup, intended use, limitations)
    - Repro Notes (env file, hardware notes, "how to rerun")
3. Reflection (800–1200 words): How the dataset's boundaries shaped the work; what you refused to include and why; what you'd change with more time.

Assessment rubric categories

- Dataset intentionality & documentation quality

- Pipeline reproducibility (versioning, scripts, configs, rerun-ability)
- Evaluation rigor (clear metrics and aesthetic evaluation criteria)
- Creative system design (mapping, temporal form, interaction/performance logic)
- Craft & iteration evidence (changelogs, experiments, meaningful revisions)
- Communication (clarity of repo, cards, and reflection)

Responsible practice checklist (consent, bias, provenance, citations)

- Consent: documented permission for any identifiable recordings; opt-out path if applicable
- Bias: describe likely skews; include at least one bias audit (e.g., distribution checks + qualitative failure review)
- Provenance: every datum traceable to source; license status recorded; no "mystery files"
- Citations: cite datasets/tools/papers; label what is yours vs. adapted; include attribution in artifact notes
- Safety & privacy (bonus but expected): remove sensitive metadata; avoid scraping private sources; secure storage plan

Assignment Prompt 2 *Instrumenting a Language Model: Real-Time Creative Coding + Guardrails + Observability*
Premise: Build a live "instrument" where a language model is not the author, but a volatile collaborator you coduct. The artistic core is how you shape, constrain, and listen to the model in real time through interface and system design.

Learning objectives

- Implement an LLM-driven interactive system with tight control of context, latency, and failure handling.
- Apply MLOps-style observability to an artistic tool (logging, eval traces, regression tests for prompts).
- Author a performance/interaction structure that foregrounds human intent, timing, and interpretive decisions.

Constraints

- You must impose hard constraints on the model's role (e.g., it can propose, but you select; it can transform, but not originate; it can speak only from a curated corpus).
- The system must expose its seams: the audience/user should perceive the control architecture (not just "a chatbot show").
- You must implement at least two guardrails (content boundaries, retrieval constraints, or output filters) and show them working.

Deliverables (artifact + docs + reflection)

1. Artifact: a live performance script (or interactive installation spec) + a demo recording (5–8 minutes).
2. System: creative coding project (e.g., Max/MSP, TouchDesigner, p5.js, Unity, etc.) that includes:
   - Context manager (prompt templates + retrieval or memory rules)
   - Latency strategy (caching, streaming, fallback behaviors)
   - Control surface (MIDI/OSC/UI) that shapes generation
3. Observability pack:
   - Logging of prompts/outputs with timestamps (redacted if needed)
   - A small suite of "behavior tests" (prompt regression cases)
   - A short "model behavior report" (what it does reliably vs. unpredictably)
4. Reflection (800–1200 words): How you balanced determinism and liveness; how failure became material; what ethical boundaries you encoded.

Assessment rubric categories (4–6)

- Interaction/performance design (form, pacing, control clarity)
- Technical robustness (latency, fallbacks, crash resistance)
- Guardrails & safety design (specific, tested, documented)
- Observability/evaluation (logs, tests, behavior report usefulness)
- Artistic authorship (how constraints create meaning)
- Documentation & usability (someone else could run/perform it)

Responsible practice checklist (consent, bias, provenance, citations)

- Consent: if using personal texts/voices as context, permissions are explicit; anonymize when needed
- Bias: identify likely stereotype traps; include at least one mitigation (prompting, retrieval, filtering, refusal)
- Provenance: document any corpora used for retrieval; show licensing/permission status
- Citations: cite model/provider/tools; cite conceptual influences; include credits in performance notes
- Transparency: disclose where the model is used, what it can't do, and what is edited/selected by you

Assignment Prompt 3 *Release Engineering for Creative ML: A Public-Facing Tool or Mini-Model*
Premise: Create a sharable "creative ML object" (a tool, library, dataset, or small model) and treat the release as part of the artwork. Your output must be usable by others, with clear boundaries, docs, and responsible distribution.

Learning objectives

- Package and release a creative ML artifact with versioning, documentation, and reproducible builds.
- Design evaluation and maintenance practices (tests, changelogs, known issues, future roadmap).
- Translate technical constraints (licensing, safety, compute limits) into aesthetic choices.

Constraints (what makes it art, not "AI content")

- Your release must embody a viewpoint: it should refuse certain uses and enable others, by design.
- You must include at least one "designed limitation" (rate limits, constrained modes, curated outputs, required citations) that shapes creative outcomes.
- No black box drop: your documentation must make the object legible as a practice.

Deliverables (artifact + docs + reflection)

1. Released artifact: one of the following

    - A small model + inference script, or
    - A dataset + loaders + example pipeline, or
    - A creative-coding extension/library that wraps ML/LLM functionality

2. Release engineering:

    - Semantic versioning + changelog
    - CI checks (lint/tests or runnable demo verification)
    - Reproducible environment (Dockerfile or env spec)
    - Minimal examples (1–2) that produce deterministic-ish outcomes

3. Reflection (800–1200 words): What you made "easy," what you made "hard," and why; how distribution changes authorship.

Assessment rubric categories (4–6)

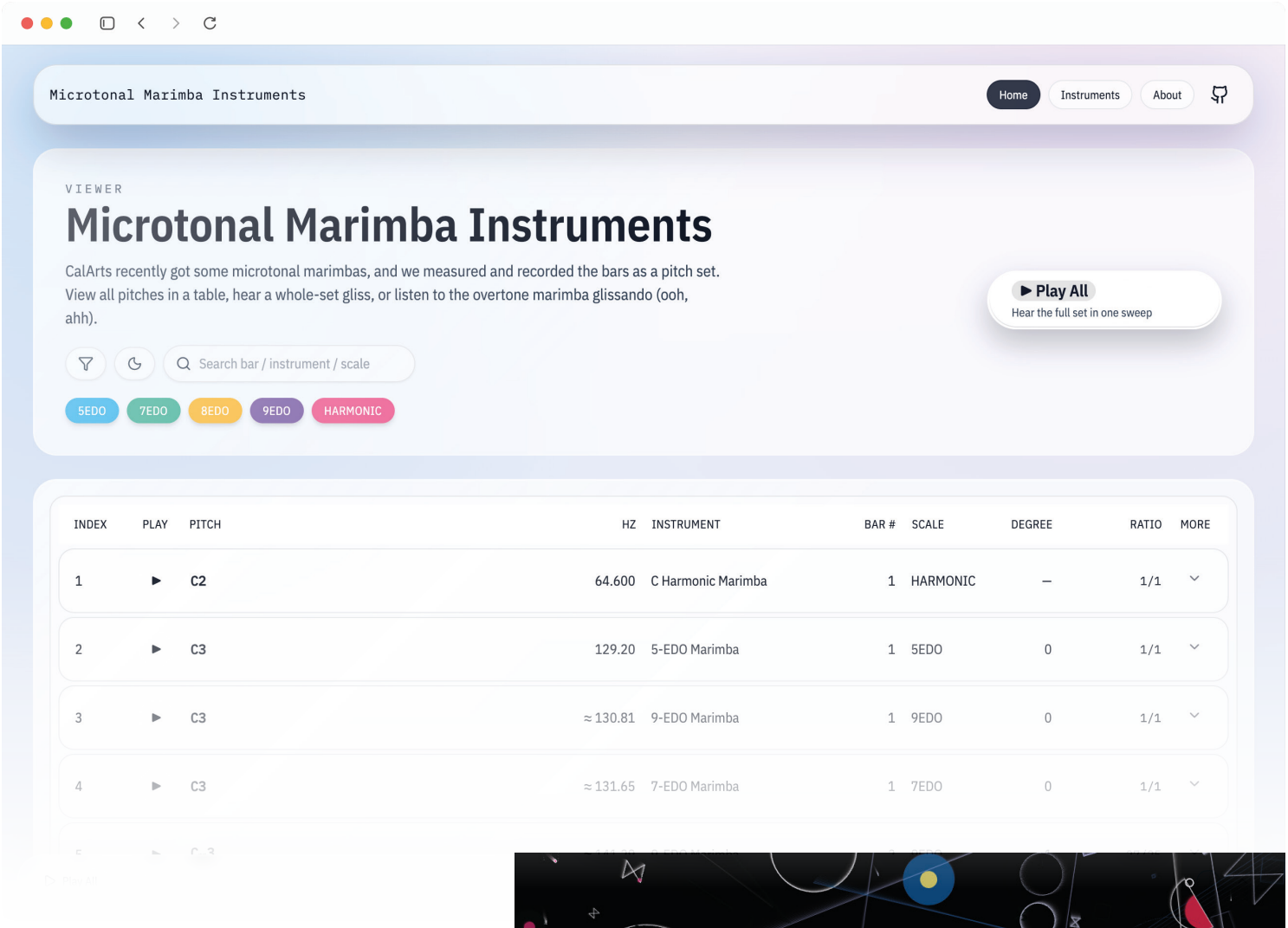
- Release quality (packaging, installability, versioning, clarity)
- Reproducibility (build/run instructions actually work)
- Evaluation & tests (meaningful checks, documented limitations)
- Responsible distribution (licenses, boundaries, safeguards)
- Artistic coherence (how constraints drive outcomes)
- Community readiness (examples, API clarity, maintenance plan)

Responsible practice checklist (consent, bias, provenance, citations)

- Consent: any included media/text has permission; no unlicensed scraped content
- Bias: include a limitations section + at least one evaluation example showing bias/failure handling
- Provenance: data/model lineage documented; training sources and transformations enumerated
- Citations: required CITATION file or credits section; license files included; attribution baked into outputs when feasible
- License & use boundaries: clearly state allowed/disallowed uses; include a short ethical use statement and reporting path

+1 (407)-922-9622 ssuarezsolis@calarts.edu

EVIDENCE



Representative work (above): **`cbassuarez/microrimba`**, a microtonal marimba library of a unique set of microtonal keyboards by Chris Banta. This was a percussion studio project I helped undertake with Tim Feeney and the percussion studio at CalArts. This kind of repository includes the exact components which might be most beneficial for a student: installing and running dependencies, creating a data model, and deploying that data model on a hosted site.

Student work, FALL 2025, untitled, description:"Interactive p5.js piece where visitors trigger evolving visuals with touch."

+1 (407)-922-9622 ssuarezsolis@calarts.edu

COURSE SNAPSHOTS

COURSE SNAPSHOT I.

Digital Instrument Building: Coding for Musicians (Hypothetical)

Level: Upper-division undergraduate / graduate
Format: In-person, lecture + lab (hybrid critique: in-room + asynchronous PR review)
Typical enrollment: 20–30
Credits: 3 (suggested)

Course Description

This course treats the digital musical instrument as a repo-based artistic system: code, sound, interaction design, documentation, version control, and performance practice. Students build instruments using contemporary creative-coding and audio ecosystems (e.g., browser-based audio, creative coding toolchains, live-coding environments, and text-based audio languages), while developing the professional competencies required to sustain large projects: modular architecture, recursive thinking, dependency awareness, licensing/ethics, testing, and collaboration.

The course culminates in a public-facing performance (in class or on campus) featuring each student's instrument and a structured technical demonstration of repository fluency.

Student Profile / Assumed Readiness

Students are majors who:

Can compose without step-by-step direction (self-directed artistic decision-making)
Have basic laptop fluency and comfort handling code (not necessarily advanced)
Are prepared to work in a shared codebase and communicate via documentation

Learning Outcomes

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

Build & ship instruments
1. Design and implement a playable digital instrument with a coherent interaction model and sound engine.
2. Integrate timing, control, mapping, and signal flow into a stable performance-ready system.

Think at project scale
3. Structure a repository for growth (clear modules, naming conventions, dependency boundaries, readable docs).
4. Use recursive models – reusable components, templates, generators, and iterative "looped" workflows (e.g., VPP-style revise cycles, codex-like write→run→refine loops, MCP-style tool protocols) – to scale from simple primitives into complex instruments.

Operate professionally
5. Use version control fluently in realistic workflows (branching, merging, reverting, rebasing, recovery strategies) and collaborate through review processes.
6. Create tests and checks appropriate to creative systems (sanity checks, performance checks, reproducibility

notes, "known failure modes") and run them through an automated pipeline when appropriate (e.g., CI checks that gate merges).

Practice critically
7. Articulate an ethical and historical framework for instrument-making (authorship, accessibility, extractive tooling, dataset provenance, bias/limitations of learning models).
8. Demonstrate awareness of the current ecosystem (libraries, toolchains, services) and communicate where the landscape is shifting – and what that means for artistic practice.

Major Assignments

1) Instrument Studies (Short Builds) — 25%

A sequence of compact instruments (1–2 weeks each) that isolate core problems:

- timing and scheduling
- mapping & gesture
- synthesis/sampling strategies
- scene/graphics pipeline integration
- failure-proofing for performance

Deliverables: repo snapshots, short demo videos or in-class demos, and brief "design notes."

2) OSS Studio: Class Repo (Collective Instrument Lab) – 20%

As a class, we will build and maintain one shared open-source repository for the entire semester – an evolving instrument ecosystem and learning artifact. We'll continually add to it together: modules, patches, examples, docs, tests, onboarding, and performance-ready "releases." Contributions happen through issues, branches, PRs, review, and merge, with a visible roadmap and release milestones.

Deliverables:

- at least one meaningful contribution per student (scoped to ability level) to the shared class repo
- PRs/issues written to clear community norms (titles, repro steps, acceptance criteria)
- basic "production hygiene" where appropriate: checks passing (CI/lint/tests), dependency notes, and changelog/release notes for milestones
- a short reflection ensuring students can explain why the change matters (musically + technically)

3) Peer Review System (PR-Based Critique) — 15%

Students participate in structured peer feedback using PRs:

- review quality is assessed (clarity, specificity, kindness, technical correctness)
- students practice "review-as-mentorship," not "review-as-judgment"

Deliverables: a minimum number of documented reviews + responses + revisions.

4) Version Control Practical (Live) — 15%

A two-part assessment:

1. Artifact: an at-home assesment, create a branch demonstrating a realistic edit cycle and recovery strategy
2. Live test: student performs a series of repo operations at the desk with instructor (screen shared to class, but command line hidden from the audience). Tools limited to command line + editor (no GUI git clients; browser allowed only for confirmation).

Evaluates: calm under pressure, correctness, recovery literacy, and ability to narrate intent.

5) Final Project: Repo-Based Instrument + Performance — 25%

Students design and build a personal instrument in a repo (choice of stack/toolchain), then perform with it (in class or campus event).

Deliverables:

- instrument repo (clean structure, README, setup instructions, dependency notes)
- performance plan (what the instrument does, what can go wrong, fallback plan)
- performance (live)
- short postmortem (what worked, what failed, what's next)

Final Project Requirements (Non-Negotiables)

- *Reproducibility: another student can set up and run a demo following your README*
- *Performance readiness: stable enough for rehearsal + live use*
- *Documented intent: your repo explains the instrument's design logic and artistic stakes*
- *Critical framing: short ethics/position statement on tooling choices (libraries, models, datasets if relevant, accessibility considerations)*

Suggested Weekly Arc (14 Weeks)

1. Instrument as system: repo-as-instrument; collaboration norms; critique protocol
2. Toolchains & environments; reproducibility; dependency reality
3. Time: clocks, scheduling, latency, determinism (creative implications)
4. Mapping: gesture → sound; constraints as expressivity
5. Synthesis/sampling strategies; designing controllable sonic behavior
6. Structure: modularity, components, and recursive patterns for scale
7. Midpoint micro-performances + refactor week (stability as artistry)
8. Graphics/space/interaction layers (visuals, sensors, interfaces)
9. Networks & collaboration patterns (shared states, ensemble instruments)
10. Testing creative systems: sanity checks, performance checks, failure modes
11. Learning models/assistants: what they're good for, what they break; ethics
12. OSS sprint: contributions + community etiquette + sustainability

COURSE SNAPSHOT II. – Major Lessons MLSN-X01-422

Course Title: Major Lessons – MLOps, Creative Coding, and Performer Practice
Course Number: MLSN-001/101/501/601-422
Credits: variable

School of: Music
Semester: Spring
Year: 2026
Day: by appointment
Time: 30-60 minutes/week (plus studio forum as scheduled)

How can you contact me?

Instructor Name: Sebastian Suarez-Solis
Preferred Name: Seb preferred, Mx. prefix is ok or none at all (i.e. do not call me Mr.)
Pronoun: they/them
Contact Email: ssuarezsolis@calarts.edu

What is this course about?

Major Lessons is individualized instruction in creative practice and professional development. In this version, the course centers composition and/or artistic production where artificial intelligence and machine learning is treated as an artistic material: not a shortcut or aesthetic preset, but a set of methods that can be designed, tested, critiqued, documented, and ethically situated.

Lessons support students in developing a body of work (scores, performances, installations, recordings, interactive systems, or hybrid forms) while building technical fluency in contemporary workflows. We treat "AI" broadly: machine learning and generative systems, but also data practice, automation, procedural design, and human-in-the-loop decision-making. Each student's "anchor practice" stays central (composition, performance, installation, design, theatre/dance media, etc.), while we also build transferable skills and critical perspectives that can travel across art forms.

This course models a studio culture of rigorous craft, reflective iteration, and responsible technology use. You will leave the term with (1) new work, (2) better process, and (3) clearer articulation of what you make, how you make it, and why it matters.

What will you learn in this course?

1. Develop an iterative artistic workflow that moves from concept → prototype → critique → revision → presentation, with documentation at each stage.
2. Integrate AI methods into an artistic practice in ways that are technically coherent, aesthetically intentional, and legible to collaborators and audiences.
3. Demonstrate craft in at least one chosen medium (score writing, sound design, interactive systems, performance materials, installation design, etc.), with attention to clarity, feasibility, and audience experience.
4. Use a critical framework to evaluate AI in the arts: authorship, labor, bias, training data, attribution, privacy, sustainability, accessibility, and cultural impact.
5. Communicate your work professionally through clear materials (scores/patches, tech riders, installation diagrams, program notes, artist statements, and portfolio documentation).
6. Collaborate effectively (when relevant), including setting roles, maintaining version control, and designing rehearsal/build timelines.
7. Reflect on process and decision-making, identifying what you will keep, discard, and develop next.

How to Succeed in This Course

Success here looks like consistency, not perfection. Bring something small each week that we can actually work on: a sketch, a test, a prompt, a patch, a page of score, a dataset idea, a rehearsal plan, a failure log. The lesson is not the performance—it's the lab.

Treat AI as a site of choices. Every system implies a set of constraints; your job is to learn what the system is doing, decide what you want, and shape results through design rather than "vibes." When you use models or tools, disclose your process, keep track of versions, and document your edits. "Show your work" is a creative advantage.

Finally: make your work shareable. Whether you're writing a score, building an installation, or training a model, your collaborators and future self should be able to understand what you did and reproduce (or responsibly remix) the setup.

How can you meet with me?

When you need to meet with me, I will be more than happy to find a time that works for the both of us. For scheduling meetings, email me at ssuarezsolis@calarts.edu and I will send you a calendar invite.

What course materials do you need for this class?

You do not need to purchase textbooks or software as a prerequisite or requirement of this course.

You will need:

- *A reliable way to take notes and keep a process log (notebook, Obsidian, Notes, etc.).*
- *A portfolio storage system (Drive folder, Git repository, or equivalent).*
- *Headphones and access to your primary creative tools (notation software, DAW, creative coding environment, etc.).*

- *For students working with AI: a plan for compute and tools (local, school lab, or cloud) appropriate to your project scope.*

The following items may be emailed to you before or during a lesson:

- Scores, recordings, technical papers, artist talks, tool documentation, and short readings on ethics and critical AI.
- Examples of artist statements, program notes, tech riders, and documentation templates.

Technology Requirements

Your exact stack depends on your practice. Common tools include:

- DAW (Ableton, Reaper, Logic, etc.) and/or notation software (Dorico/Sibelius/MuseScore)
- Max/MSP, Pure Data, SuperCollider, or similar
- Creative coding: Python, strudel, p5.js, openFrameworks, TouchDesigner, Unity/Unreal (as relevant)
- Versioning and sharing: Drive + clear folder conventions, or Git for code-based work
- AI workflows may include: prompt-based tools, model-assisted editing, open-source generative audio/image/video libraries, or custom pipelines. We will prioritize transparent, reproducible workflows over mystery-box results.

How Will Your Learning Be Assessed?

This is an individual-instruction course; assessment is based on consistent engagement, documented progress, and completion of agreed-upon deliverables.
Your final grade will be assessed through completion and quality of process (not "talent"):

Assignment	% of Grade
Weekly Lesson Preparedness	25%
Iteration & Documentation	25%
Portfolio Deliverables	40%
Studio Citizenship	10%

CalArts' grading system is High Pass (HP), Pass (P), Low Pass (LP), and No Credit/Fail (NC). More info about CalArts' grading system here.

Late Policy

Major Lessons runs on momentum. If something delays your work, email me early with (1) what's blocking you, (2) what you can still deliver this week (even small), and (3) what support you need. We will renegotiate scope rather than pretend time doesn't exist.

Course Assignments

There is one final assignment for this course, and here's the breakdown:

Final Portfolio Package (Weeks 12-13 / Finals)
A polished submission appropriate to your medium, including:

- Final work (or best-possible state if iterative project)
- Documentation (photos/video, score/patch/code, diagrams)
- Artist statement or program note (300-600 words)
- Process reflection (1-2 pages): what you learned, what you'd redo, and what's next

Feedback Model

Feedback Model (How critique works here)
We will use a dialogic feedback approach. In lessons and forums, feedback will prioritize:

- What the work is doing (observations)
- What the artist wants (intent)
- What options exist next (offers)

Traditional "taste-based verdicts" are less useful than actionable choices. You are encouraged to request specific kinds of feedback ("timing," "form," "install flow," "model behavior," "notation clarity," etc.).

What do I need to know about academic honesty?

CalArts is a community of artists. In this community, all members including faculty, staff, and students are responsible for maintaining standards of academic and artistic honesty. As a student and a member of the CalArts community, you are here to get an education and are, therefore, expected to demonstrate integrity in your academic and artistic endeavors. You are evaluated on your own merits. Cheating, plagiarism, fabrication, or other kinds of academic dishonesty are considered unacceptable behavior and will result in formal disciplinary action, as determined by the faculty member, the dean of the student's school and the Office of the Provost.

Plagiarism is the use of ideas and/or quotations (from the Internet, books, films, television, newspapers, articles, the work of other students, works of art, media, etc.) without proper credit to the author/artist. While the argument in a paper can be enhanced by research, you are cautioned to delineate clearly your own original ideas from source material. To do so, CalArts recommends introducing your source material (either quoted or paraphrased), noting when the source material ends, and providing citations for source materials using standard documentation formats. Misrepresentation of source material as your own original work and failing to credit it is plagiarism. If you have questions regarding plagiarism or

would like direction on how to credit source material, please contact one of the CalArts reference librarians for more information.

How do I obtain reasonable accommodations for this course?

CalArts is committed to providing reasonable accommodations in compliance with ADA of 1990 and Section 504 of the Rehabilitation Act of 1973 to students with documented disabilities or otherwise documented. If you are requesting accommodations for equal access to this course, please register with the Disability Services Office, for the facilitation and verification of need. The Disability Services Office will meet with you and communicate with your faculty about appropriate and reasonable classroom accommodations. You are encouraged to use these procedures early in the semester, so that the proper arrangements can be in place throughout this course. The Disability Services Office is located in the Health & Wellness Office in F201, and can be contacted by emailing DSO@calarts.edu. Check out Disability Services for more information.

How do I learn more about sexual respect resources?

CalArts is committed to treating all members of the community with dignity, empathy, and respect. Accordingly, the Institute prohibits the following forms of sex-based misconduct: sexual assault, sexual harassment, gender-based harassment, dating violence, domestic violence, sexual exploitation, and stalking.

All faculty, staff, teaching assistants with teaching responsibilities are a non-confidential resource and have an obligation to report any information about gender- or sex-based discrimination or harassment to the Title IX coordinator. This includes all incidents of alleged sexual or dating violence. The Title IX coordinator is responsible for investigating violations of the sexual misconduct policy. Go here for more information about the CalArts Prohibited Discrimination, Harassment, and Sexual Misconduct Policy.

Members of the CalArts community who believe that they have been subjected to Sex-Based Misconduct are encouraged to report such incidents to the Institute and, where applicable, to local law enforcement. Confidential resources include the Institute Counselors and the Associate Vice President for Student Experience, Health and Wellness.

Resources for Support & Learning

Wellbeing Support
Recent nationwide surveys of college students consistently find that stress, sleep problems, anxiety, depression, interpersonal concerns, death of a significant other and alcohol use are among the top ten health impediments to academic performance. Students experiencing personal problems or situational crises are encouraged to check out CalArts' mental health services.

Need help right away? The crisis counseling line dedicated to CalArts students only is free and can be called from anywhere in the U.S., 24 hours a day, 7 days a week at (855) 364-7981.

Basic Needs Support

If you face challenges securing food or hygiene products, you are not alone, and CalArts can help during this time of crisis. We invite you to learn about many resources available to support you through CalArts' Basic Needs Center.

Writing Support
One-on-one trained writing tutors are ready to work on any writing-related task with you in the Writing Center. Writing-related tasks include class papers, artist statements, grant applications, resumes, etc. For more information or to book an appointment, visit the Writing Center.

English Language Support
Come attend a workshop! The Writing Center hosts biweekly ELL/language support workshops for MFA students, covering a range of English-language related skills and strategies. Check out the Writing Center!

Library
Need help finding a book or other materials, researching sources for a paper or project, or just need a little guidance in the library? There is a CalArts Librarian that specializes in the resources and search strategies specific to each School and métier. The following are just a few of the ways in which the Library can help you:

- Book an appointment to meet with a librarian about your research needs!
- Print a paper or project.
- Borrow a laptop.
- Use the group study room or computer lab.

For more information or to learn about other resources for support, check out https://calarts.edu/life-at-calarts/support-and-advocacy

Please keep this syllabus easily accessible so that you can refer to it throughout the semester. I look forward to getting to know you and supporting your learning in this course.

+1 (407)-922-9622 ssuarezsolis@calarts.edu

ON INCLUSION *My teaching and creative research start from a practical belief: technology only matters in the arts when it becomes shareable, when students can access it, understand it, adapt it, and critique it. Because AI tools can amplify both possibility and inequity, inclusion and ethics are not add-ons to my pedagogy; they are design constraints that shape how I structure assignments, choose toolchains, run critique, and evaluate work. In a studio where students arrive with uneven prior experience, uneven equipment, and uneven time to experiment, my goal is to make participation predictable and achievable while asking students to think rigorously about authorship, data rights, bias, and responsible disclosure.*

At the beginning of a course, I set "accessibility defaults" that prevent barriers from becoming the hidden curriculum. That begins with a simple promise: every assignment has a minimum viable pathway that can be completed without specialized hardware, a paid software stack, or a dedicated GPU. When advanced compute is useful, it appears as an optional tier or a shared resource—never as the condition for meeting core learning outcomes. I separate assignments into (1) the concept students must demonstrate, (2) multiple implementation routes to get there, and (3) stretch goals for students who want to pursue advanced workflows. This approach makes the studio legible: students can choose a pathway that fits their circumstances without the work being framed as "lesser." It also supports cross-disciplinary cohorts (musicians, designers, performers, and visual artists) because the emphasis is on transferable concepts (signal flow, procedural thinking, evaluation and iteration) rather than a single "correct" tool.

To reduce barriers further, I design assessment around evidence of process rather than production polish. Students can show learning through multiple modalities: an artifact (score, recording, installation documentation, performance video), a short reflection, a process log (screenshots, patch versions, prompt and parameter logs, dataset notes), and a brief walkthrough (live or recorded). Rubrics reward clarity of intent, coherence of method, iterative development, and critical reflection –criteria that do not correlate with access to expensive tools. I also build "time equity" into the course: in-class build sessions, structured peer support, and office hours that include technical onboarding, not only conceptual advising. When needed, asynchronous critique options (recorded presentations, written responses) ensure that work schedules, caregiving responsibilities, disability, or time zones do not determine who can be meaningfully present.

Critique itself is a major site of inclusion. I use dialogic critique protocols (including structured response processes) to shift feedback away from performative confidence and toward attentive description, artist intent, and actionable inquiry. Students learn to ask: What is this work trying to do? What choices support that intention? What alternatives are available? This matters in AI-mediated practice because it keeps the focus on method and responsibility. My aim is to make critique a place where students can safely articulate uncertainty, test ideas, and learn the vocabulary of the field without being punished for not already having it.

Within that inclusive studio culture, ethics becomes a daily practice rather than a one-off lecture. AI-based art making introduces risks at multiple levels: data rights, privacy, authorship, bias, and the temptation to over-trust outputs; as such, I treat ethical decision-making as part of craft. Students are asked to document the materials and systems they use (datasets, models, prompt strategies, post-processing) and to treat transparency as a professional norm. "Process & Tools" notes accompany every submission. This is not punitive; it builds a shared language for critique and helps students understand their work as a set of accountable choices.

To make that concrete, I build assignments where ethics and AI technique are inseparable. One early example is a text-to-audio prototyping lab that treats generative output as sketch material rather than an endpoint. Students generate short sonic studies from text prompts, then analyze what the system gave them (timbre, dynamics, spectral density, phrasing) and rebuild or transform those ideas using accessible tools: DAWs, basic synthesis, or patch-based environments. The learning goal is rapid ideation paired with interpretive control: students practice moving

from "generated suggestion" to intentional composition, while also documenting the prompt history, model used, and any post-processing. We use this assignment to discuss what counts as authorship, what "disclosure" should look like in program notes, and how data provenance shapes aesthetic and ethical choices.

A second example centers on machine listening as a compositional partner, which is especially relevant for installation and performance contexts. Students build a simple listening pipeline – feature extraction (onsets, loudness, spectral centroid, MFCCs), segmentation, or classification – then map those features to sound synthesis, visual parameters, or performative cues. The focus is not on building a "perfect" model; it is on understanding what the system can and cannot "hear," and how those limitations become artistic material. Importantly, this assignment is designed to run on modest hardware using open tools, and it invites critical questions: What gets recognized, what gets ignored, and why? How do microphone choice, environment noise, and dataset selection shape outcomes? Students learn that "AI perception" is always a design choice, and that design has consequences.

A third example addresses the politics of training data through a dataset critique and generative structures project. Students assemble a small, consent-forward dataset (self-recorded sounds, public-domain texts, properly licensed materials, or documented community-shared sources), and they create a short "dataset card" describing provenance, permissions, exclusions, and potential risks. They then implement a generative structure appropriate to the medium with rules-based systems, Markov chains, small sequence models, or constraint-based generation so that students can feel how aesthetics emerge from data and structure. Critique includes an "impact pass": who benefits from this work, who could be misrepresented, and what context is needed for responsible interpretation. This turns "ethics" into something students can point to in their process and evaluate in each other's work, rather than an abstract posture.

Across these projects, I maintain a few consistent studio norms. First, consent and rights are non-negotiable: students are expected to use self-generated, public-domain, or properly licensed materials, and projects involving identifiable people (voice, image, text tied to individuals) require explicit consent and a clear plan for privacy-preserving practice. Second, we treat generated outputs as non-authoritative: they are not evidence, and they do not replace research or verification when a project references history, communities, or claims about identity. Third, we make space for failure and bias as legitimate critique material, where hallucination, mode collapse, stereotype artifacts, or misclassification are analyzed as part of the work's method and meaning.

Finally, before students share AI-mediated work publicly (festival submissions, online release, press), I use a lightweight "risk review" checkpoint. Students confirm rights and consent, include appropriate attribution and disclosure, consider plausible misuse (e.g., impersonation via voice), and define mitigation steps (framing text, captions, limited distribution, or substituting materials). This checklist functions like a studio pre-flight: fast, concrete, and normal, an expectation of professional practice.

In the context of the UT Austin College of Fine Arts, where AI is increasingly present across visual art, design, performance, and installation, this framework lets me teach advanced technical concepts while protecting access, agency, and responsibility. It ensures that students are not sorted by privilege or prior exposure, and it builds a studio culture where experimentation is paired with accountability: students learn not only how to use powerful tools, but also how to explain, critique, and ethically situate the work they make with them.