MEDIART 206 / COMPSCI 208: FALL 2025 - SESSION 2

Computer Graphics: Code as Artistic Medium

Course Dates and Meeting Time: 2025/10/20 – 2025/12/11 - Tuesday and Thursday from 14:45 to 17:15
Exam Week: 2025/12/09 – Tuesday – 12:00 – 15:00
Room: IB 2025
Academic Credit: 4 Credits
Course Format: Seminar
Office Hours: TBD

Instructor's Information

Benjamin L Bacon
Associate Professor of Media and Arts
Email: benjamin.bacon@dukekunshan.edu.cn
Office: IB 1027
Online: https://scholars.duke.edu/person/benjamin.bacon; https://benjaminbacon.studio

Benjamin Bacon is an Associate Professor of Media & Arts and the Major Convener of the Computation & Design program at Duke Kunshan University (DKU). He is the co-director of the Duke-DKU Presence Lab and the co-founder of the Design, Technology, and Radical Media Lab (DTRM). He is a fellow at V2_Lab for the Unstable Media and the co-lead of the Art, Media, and Cybernetics (AMC) working group at the American Society of Cybernetics (ASC). His practice centers around explorations into computation, its qualities and characteristics as a creative medium, and its changing relationship with society and industry perception. Four main exploratory trajectories have persisted in his research and practice: computational media, machine art, sound, and reality media. His creations have taken the form of mechanical sculptures, machine-learning neural networks, networked systems, experimental interfaces, body-hacking, and sound. His methodology as an artist is fundamentally rooted in the design research process. It is experimental in its essence, often reliant on direct interaction with materials. His conceptual approach is at times playful, at times critical, at times commentary, and at times speculative.

What is this course about?

Computer Graphics: Code as Artistic Medium examines the design and development of computer graphics through the lenses of **creative coding, generative design, computer-aided composition,** and **animation.** Using the open-source **Processing IDE**, students will learn foundational programming concepts—variables, loops, conditionals, data structures, and object-oriented design—through hands-on projects that merge art and computation.Throughout the course, we approach code not only as a technical skill but as an **artistic material** and a **conceptual process**. Through a balance of lectures, readings, studio practice, and critical writing, students will learn to generate and manipulate visual systems programmatically, explore historical and theoretical approaches to algorithmic and rule-based art and develop individual and collaborative practices grounded in experimentation, iteration, and reflection. Each week introduces a new framework-form, repetition, transformation, parameterization, visualization, and simulation—that builds toward a final curated portfolio of computational artworks. By the end of the course, students will be able to design autonomous generative systems, visualize data creatively, and critically evaluate the cultural and aesthetic implications of computer-generated form.

What background knowledge do I need before taking this course?

No prior programming experience is required. The course begins with an introduction to coding concepts through the Processing IDE, designed for artists and designers.

Students should have: Basic digital literacy (file management, use of creative software, etc.). An interest in visual design, art, and computational processes. Willingness to experiment, iterate, and engage in both technical and conceptual work. Experience in any of the following areas can be helpful but is **not required**: Digital media, photography, or visual arts. Basic mathematics or geometry. Design, architecture, or interactive media.

All technical skills will be developed in class through lectures, demonstrations, and guided studio projects.

What will I learn in this course?

After successfully completing this course, students will be able to:

1. **Apply** creative coding techniques to generate visual form and behavior.
2. **Design** and **implement** rule-based and parametric systems that exhibit variation, autonomy, and emergence.
3. **Synthesize** theoretical and historical readings on code-based art into critical reflection and creative decisions.
4. **Communicate** ideas effectively through visual composition, documentation, and critique.
5. **Evaluate** generative systems critically—identifying authorship, bias, and conceptual intent in computational artworks.
6. **Curate** and **present** a cohesive digital portfolio that demonstrates technical proficiency, conceptual depth, and reflective insight.

What will I do in this course?

The course combines lecture, discussion, studio, and reflective writing to integrate creative coding practice with critical and theoretical study. Each week introduces new technical and conceptual methods that build toward a portfolio of computational artworks. Assignments are scaffolded to develop both technical proficiency and reflective insight, with opportunities for feedback and revision at every stage.

Course Structure and Format

The structure of most of the 2.5-hour course sessions each week is as follows:

- **Day 1 (2.5h): Conceptual framing + theory + technical demonstrations.**
  - 50% Lecture (≈75 min)
  - 10% Discussion (≈15 min)
  - 40% Coding Lab (≈60 min)

- **Day 2 (2.5h): Project development, one-on-one feedback, peer critique.**
  - 100% Studio / Critique

Pre-Class Reading & Reflection Activities

- **Description:** Students complete assigned readings before each week's lecture and studio session. They annotate texts, highlight key ideas, and write short reflections (350–700 words) connecting readings to ongoing projects. Readings focus on code as an artistic medium, rule-based aesthetics, and systems thinking in art and design.
- **Learning Objective Alignment:** Supports Learning Outcomes 1, 3, and 5: Understand and apply conceptual frameworks in creative coding. Synthesize theoretical and historical readings into critical reflection. Evaluate authorship, bias, and conceptual intent in computational artworks.

Lectures & Demonstrations

- **Description:** Each lecture introduces the historical, theoretical, and technical foundations for that week's project. Lectures are supported by live coding demonstrations in the Processing IDE, artist case studies (e.g., Molnár, LeWitt, Reas, McCarthy), and short analytical discussions that translate theory into practice.

- **Learning Objective Alignment:** Supports Learning Outcomes 1, 2, and 3. Apply creative coding techniques to generate visual form and behavior. Design and implement rule-based and parametric systems. Integrate theoretical frameworks into coding practice.

In-Class Discussions

- **Description:** Following each lecture, students participate in guided small-group and full-class discussions that connect weekly readings to coding practices and project development. Each week, **rotating "student discussion facilitators"** will help summarize key ideas from the assigned readings, introduce one or two guiding questions, and lead 10–15 minutes of discussion. Facilitators are expected to prepare in advance by reviewing the week's texts and identifying connections between theoretical ideas and creative coding methods. This activity emphasizes critical thinking, verbal articulation, and collaborative learning.
- **Learning Objective Alignment:** Supports Learning Outcomes 3, 4, and 5. Synthesize theoretical readings with personal creative practice. Communicate ideas effectively through dialogue and critique. Evaluate artistic and conceptual intent in computational systems.

Coding Labs

- **Description:** Hands-on labs (approximately 40% of Day 1) teach programming fundamentals through guided exercises. Students experiment with new techniques—loops, functions, data handling, simulation—and immediately apply them in class. Labs culminate in iterative sketches that serve as starting points for weekly projects.
- **Learning Objective Alignment:** Supports Learning Outcomes 1 and 2. Apply creative coding to produce dynamic, rule-based form. Implement parametric and generative logic in Processing.

Studio Sessions (Day 2 Each Week)

- **Description:** Studio sessions provide extended project time for experimentation, iteration, and individualized feedback. Students develop weekly projects, refine aesthetics and code, and present iterations for informal critique. The studio functions as both a collaborative learning environment and a production lab.
- **Learning Objective Alignment:** Supports Learning Outcomes 1, 2, 4, and 5. Apply creative coding methods to generative systems. Communicate ideas effectively through documentation and critique. Evaluate artistic intent and system behavior through iteration and feedback.

Weekly Projects (1–6)

- **Description:** Six cumulative creative coding projects form the core of the course. Each project includes code, documentation, visual exports, and a reflective writing component:
  1. **Procedural Poster** – form as process
  2. **Two Rule-Sets Pattern** – repetition and instruction
  3. **Image Re-Written** – transformation and translation
  4. **Composition System (x16)** – parameterization and variation
  5. **Poetic Data Graphic** – data as aesthetic material
  6. **Tiny Ecosystem** – simulation and emergence

- **Learning Objective Alignment:** Supports Learning Outcomes 1, 2, 3, and 4. Apply creative coding and system design techniques. Synthesize theory and practice through written reflection. Communicate visually and conceptually through iterative design.

Peer Critiques & Feedback Rounds

- **Description:** Students present work-in-progress and receive peer feedback weekly. Critiques emphasize constructive analysis of concept, process, and technical structure. Students learn to articulate and evaluate aesthetic decisions in computational form. Peer review is also incorporated into the final portfolio and essay.
- **Learning Objective Alignment:** Supports Learning Outcomes 4 and 5. Communicate and defend artistic and technical choices. Evaluate computational systems critically through observation and dialogue.

Final Portfolio & Reflective Essay

- **Description:** In Week 7, students compile all six projects into a cohesive portfolio, accompanied by a 1,000–1,200 word essay. The essay situates their body of work within the theoretical frameworks studied throughout the course. Portfolios are presented during the final critique session and assessed for coherence, technical quality, and critical depth.
- **Learning Objective Alignment:** Supports Learning Outcomes 3, 4, 5, and 6. Synthesize theoretical, aesthetic, and technical dimensions of practice. Communicate ideas effectively through presentation and writing. Curate and reflect upon personal growth as a computational artist.

How can I prepare for the class sessions to be successful?

To be successful in this course, students should come to each class prepared to engage both conceptually and technically. Before the first session of each week, **complete the assigned readings and reflections**, *noting connections between the theoretical ideas and your ongoing project*. **Review any sample code or demonstrations posted on Canvas and be ready to discuss them during class**. Each week builds on previous material, so maintaining steady progress is essential. During studio sessions, bring your laptop with Processing installed, arrive with your current project files, and be prepared to code, debug, and share your work. Outside of class, plan to **dedicate several hours** to developing projects, refining code, and revising reflections. ***Students serving as discussion facilitators should be ready to summarize key readings and pose thoughtful questions.*** Above all, treat coding as a creative and iterative process—experiment freely, document your development carefully, and connect your practice to the ideas explored in readings and discussion.

What texts, materials, and equipment will I need?

All readings and resources for this course are freely accessible online or through the Duke Kunshan Library. The primary textbook is *Form + Code in Design, Art, and Architecture* by Casey Reas and Chandler McWilliams (Princeton Architectural Press, 2009), available digitally via the Internet Archive. Additional readings include short essays and excerpts by artists and theorists such as John Maeda, Vera Molnár, Sol LeWitt, Frieder Nake, Nick Montfort, Philip Galanter, Margaret Boden, and others. All required links and PDFs are provided on the course Canvas site and can be accessed electronically without purchase.

Students will use the Processing IDE (free download at https://processing.org) for all coding exercises and projects. A laptop capable of running Processing is required for every class. No additional equipment or paid software is necessary.

Required Texts & Readings

(All URLs verified October 2025. Most are open access or library-available. Almost all texts are uploaded to Canvas except John Maeda, Design by Numbers (use the URL))

1. Casey Reas & Chandler McWilliams, *Form + Code in Design, Art and Architecture* (2009).
   - https://archive.org/details/formcodeindesign0000reas
2. John Maeda, *Design by Numbers* (MIT Press, 1999).
   - https://archive.org/details/designbynumbers0000maed
3. Sol LeWitt, “Paragraphs on Conceptual Art.” *Artforum* (1967).

- https://monoskop.org/images/3/3d/LeWitt_Sol_1967_1999_Paragraphs_on_Conceptual_Art.pdf

4. Frieder Nake, “There Should Be No Computer Art.” (1971).
   - https://dam.org/museum/wp-content/uploads/2021/05/Nake1971-there-should-be-no-computer-art.pdf
   - https://dam.org/museum/essays_ui/essays/there-should-be-no-computer-art/
5. Aline Guillermet, “Vera Molnar’s Computer Paintings.” *Representations*, no. 149, 2020, pp. 1–30.
   - https://www.jstor.org/stable/26908911
6. Vera Molnár, “Toward Aesthetic Guidelines for Paintings with the Aid of a Computer” (1975).
   - https://www.jstor.org/stable/1573236
7. Nick Montfort et al., *10 PRINT CHR$(205.5 + RND(1)); : GOTO 10* (MIT Press, 2013).
   - https://10print.org
   - https://10print.org/10_PRINT_121114.pdf
8. Philip Galanter, “What Is Generative Art? Complexity Theory as Context.” (2003).
   - https://philipgalanter.com/downloads/ ga2003_paper.pdf
9. Margaret A. Boden & Ernest A. Edmonds, “What Is Generative Art?” *Digital Creativity* 20 (1–2) (2009).
   - https://creativecoding.soe.ucsc.edu/courses/cmpm202_w20/texts/Boden_Edmonds_WhatIsGenerativeArt.pdf
10. Harold Cohen, “The Further Exploits of AARON, Painter.” (1995).
    - https://www.kurzweilcyberart.com/aaron/pdf/furtherexploits.pdf
11. Golan Levin**,** “Some Notes on Visualization Without Computers.” *Sonic Acts* Conference (2006).
    - https://www.flong.com/archive/texts/essays/essay_sonicacts_2006/index.html
12. Lauren McCarthy, Project statements (*p5.js*, *Follower*, *Social Turkers*).
    - https://get-lauren.net/p5-js
    - https://get-lauren.net/Follower
    - https://get-lauren.net/Social-Turkers
13. Kate Compton, “Casual Creators.” *Proceedings of ICCC 2015*.
    https://computationalcreativity.net/iccc2015/proceedings/10_2Compton.pdf

Supplementary and Online References

The following open-access online resources provide tutorials, documentation, and examples that support technical and conceptual learning in the course. Students are encouraged to explore them throughout the semester for reference and inspiration.

1. Processing.org — https://processing.org
   The official website for the Processing IDE, featuring reference documentation, code examples, and community libraries. Essential for installation, syntax lookup, and exploring generative art projects.
2. Learning Processing — http://learningprocessing.com
   Companion website to Daniel Shiffman’s textbook *Learning Processing*, offering video tutorials, exercises, and example code for beginners. A practical resource for reinforcing programming fundamentals and creative coding techniques.
3. The Nature of Code — https://natureofcode.com
   A comprehensive, project-based guide by Daniel Shiffman that explores physics simulations, autonomous agents, and emergent systems in creative coding. Directly relevant to Weeks 5–6 (*Data as Material* and *Simulation & Emergence*).

How will my grade be determined?

Graded Assessments and Grading Breakdown

Assessment in this course balances creative production, theoretical engagement, and reflective practice. Grades reflect a student’s ability to integrate conceptual understanding, technical execution, and critical reflection through iterative projects and discussions. Each assessment component contributes to the achievement of the course learning outcomes (see rubrics for details).

Grade Breakdown

Component	Description	%
Projects 1–6	Six creative coding projects developed through weekly lectures and studio sessions. Each project includes code, documentation, and a reflection linking technical methods to conceptual ideas.	**55%**
Reading Reflections	Six short written responses (350–700 words each) connecting weekly readings to creative process and code decisions.	**20%**
Discussion Facilitation & Participation	Rotating student facilitators summarize weekly readings, pose questions, and lead 10–15 minute discussions. All students are assessed for participation in critiques, feedback, and studio collaboration.	**10%**
Final Portfolio & Reflective Essay	Curated collection of all six projects, presented with documentation and a 1,000–1,200 word essay synthesizing readings and creative practice.	**15%**
Total		**100%**

Please refer to the following scale for the final grade. **Note:** Grades are not rounded up at cutoff thresholds. For example, a 97.5% will be recorded as a 97% (A), not an A+.

A+= 98% - 100% **A** = 97% - 93%; **A-** = 90% - 92%; **B+** = 87% - 89%; **B** = 83% - 86%; **B-** = 80% - 82%; **C+** = 77% - 79%; **C** = 73% - 76%; **C-** = 70% - 72%; **D+** = 67% - 69%; **D** = 63% - 66%; **D-** = 60% - 62% **F** = 59% and below

Graded Assessment Components and Rubrics

1. Weekly Reading Reflections – 20%

- **Description:** Students submit a short reflection (350–700 words) that synthesizes weekly readings, coding labs, and discussions. Reflections should not summarize but rather critically engage with how theoretical frameworks and artistic precedents inform each student's creative process.
- **How this supports learning:** Encourages students to connect conceptual readings with their own technical and aesthetic choices in Processing. Supports Learning Outcomes 1, 3, and 5 (apply, synthesize, evaluate).
- **Structure:** Graded on conceptual understanding, integration of ideas, originality, and clarity.

Criteria	Excellent (Full)	Proficient (3/4)	Developing (2/4)	Needs Improvement (1/4)
Conceptual Understanding	Demonstrates deep insight into readings; articulates clear synthesis of theory and practice.	References key concepts accurately with some synthesis.	Mentions readings but lacks analysis or precision.	Minimal or incorrect understanding of key ideas.
Integration with Practice	Links theory directly to project development; cites specific examples.	Connects reading to practice in general terms.	Mentions practice but with vague or weak connection.	No clear relationship between theory and work.
Insight & Originality	Raises new questions or perspectives; shows critical independence.	Engages thoughtfully but predictably.	Limited or surface-level reflection.	Lacks reflection or insight.
Clarity & Evidence	Well-structured, polished writing; includes examples or documentation.	Mostly clear writing with minor issues.	Unfocused or poorly edited; weak examples.	Disorganized or missing supporting detail.

2. Creative Coding Projects (1–6) – 55%

- **Description:** Each project demonstrates an evolving synthesis of conceptual understanding, algorithmic structure, and aesthetic decision-making. Students will produce a Processing sketch, document their process, and submit a short reflection.
- **How this supports learning:** Promotes technical mastery, iterative development, and design thinking through experimentation and critical reflection. Supports Learning Outcomes 1, 2, and 4.
- **Structure:** Graded on concept, execution, design, and documentation.

Criteria	Excellent (Full)	Proficient (3/4)	Developing (2/4)	Needs Improvement (1/4)
Concept & Intent	Clear, original, and conceptually rich; grounded in course ideas.	Solid concept; shows understanding of assignment goals.	Concept unclear or underdeveloped.	Minimal concept or unrelated to brief.
Technical Execution	Code is efficient, functional, and demonstrates mastery of Processing.	Code functions with minor issues; shows good understanding.	Code partially functional or lacks clarity.	Code incomplete or nonfunctional.
Aesthetic & Design Quality	Strong visual composition and cohesion; demonstrates deliberate choices.	Visually effective with some inconsistencies.	Basic or uneven design decisions.	Weak or missing visual structure.
Process Documentation	Thorough documentation and iteration evidence; clear reflection.	Includes documentation with minor gaps.	Partial documentation or unclear iteration.	Missing or disorganized process record.

3. Discussion Facilitation & Participation – 10%

- **Description:** Each week, one or two students act as *discussion facilitators*, summarizing key readings and posing critical questions to guide conversation. All students are expected to contribute to discussions and critiques.
- **How this supports learning:** Develops communication, synthesis, and leadership skills. Reinforces conceptual understanding through dialogue. Supports Learning Outcomes 3, 4, and 5.
- **Structure:** Assessed on preparation, facilitation, connection to course content, and collaboration.

Criteria	Excellent (Full)	Proficient (3/4)	Developing (2/4)	Needs Improvement (1/4)
Preparation	Demonstrates comprehensive understanding of readings; arrives with notes and guiding questions.	Good preparation with accurate summaries.	Some preparation but lacks depth or focus.	Unprepared or inaccurate summary.
Facilitation Quality	Leads dynamic, inclusive discussion; encourages multiple viewpoints.	Guides discussion effectively with some support.	Limited engagement or unclear direction.	Minimal effort; does not facilitate.
Critical Connection	Draws meaningful links between readings and projects.	Connects to projects generally.	Weak or tangential connections.	No link to course content.
Collaboration	Actively includes peers; models respect and curiosity.	Participates constructively.	Passive participation.	Disengaged or disruptive.

4. Final Portfolio & Reflective Essay – 15%

- **Description:** The final portfolio compiles all six projects and process documentation. Accompanied by a 1,000–1,200 word reflective essay connecting the student's artistic development to course texts and methods.
- **How this supports learning:** Encourages synthesis of conceptual, aesthetic, and technical growth. Supports Learning Outcomes 3, 4, 5, and 6.
- **Structure:** Evaluated on presentation, coherence, reflection, and writing quality.

Criteria	Excellent (Full)	Proficient (3/4)	Developing (2/4)	Needs Improvement (1/4)
Curation & Presentation	Cohesive, well-organized presentation; clear design identity.	Organized presentation with minor inconsistencies.	Some organization but weak curation.	Disorganized or incomplete.
Critical Reflection	Insightful synthesis of readings and projects; strong self-analysis.	Connects ideas with adequate reflection.	Limited or descriptive reflection.	Lacks synthesis or depth.
Technical & Visual Quality	Polished final projects; demonstrates refinement and evolution.	Completed projects; shows progress.	Some incomplete or underdeveloped work.	Missing or unrefined projects.
Writing Mechanics	Well-written, clear, and analytical.	Mostly clear with few errors.	Unclear or superficial writing.	Poorly structured or incoherent.

What are the course policies?

To maintain a productive, respectful, and equitable learning environment, all students are expected to adhere to the following course policies:

Course Format

- This course is taught **in person only** and is not available in an online or hybrid format.
- **Accommodations** for remote participation or alternative formats will only be granted with **written approval from the university.**

Attendance Policy

- **Attendance is required.** Active participation in discussions, studio, and group activities is central to your success in this course.
- **Excused absences** follow university policy. If you need to miss class for an excused reason (e.g., illness, religious observance, university-sanctioned events), **you must notify the instructor in writing prior to the start of the class session** and provide appropriate documentation when required.
- You are allowed **two unexcused absences** without penalty. Beginning with the **third unexcused absence**, each additional absence will result in a **5% deduction from your final course grade**.
- **Arriving more than 10 minutes late** to class will be recorded as an absence unless previously approved by the instructor.
- An **attendance sheet** will be used in each session. You are responsible for signing in.

Technology Policy

- **Laptops, tablets, and mobile phones are not allowed during lectures and discussions.** This is to support focus, deep reading, and meaningful engagement with peers and ideas.
- Devices are permitted during **studio sessions and assignment work time**, where digital tools are required.

- If you have accessibility needs that require device use during lecture/discussion, please speak with the instructor in advance.

Assignment Submission and Late Work

- All assignments must be submitted through **Canvas** by the posted deadline.
- **Submit each Processing project as a compressed .zip folder containing the .pde source code, exported visuals (.png, .jpg, or .mp4), and a short README or process documentation file (.pdf or .md).**
- Written reflections and the final reflective essay should be uploaded as **.pdf** or **.docx** files.
- **Each submission must include all required files for review:** code, visuals, and written components. Missing elements may result in point deductions.
- **Late work will receive a 15% deduction per 24 hours past the deadline.** Assignments more than **three days late will not be accepted** without a documented emergency or a pre-approved extension.
- If you are uncertain whether a particular tool, code library, dataset, or generative approach is permitted for an assignment, consult the instructor before submission.
- To protect your work, **keep versioned backups of your Processing sketches and exported media.** Screenshots or commit logs documenting progress are recommended for verification in case of technical failure.

Academic Integrity & AI Use

- Unless **explicitly permitted** in the assignment prompt, the use of **generative AI tools** (e.g., ChatGPT, Claude, Gemini) to code, write, paraphrase, or revise any portion of your assignments is **strictly prohibited**.
- If AI use is allowed for a specific assignment, that will be clearly stated in the prompt.
- Using AI when not permitted will result in a **grade of 0** for the assignment.
- A **second violation** will be treated as an **academic integrity violation** and reported according to university policy.
- **Permitted Tools:** You may use Grammarly or other grammar/spell-checkers for surface-level corrections.
- If you're ever unsure whether a tool or platform is acceptable, **ask the instructor in advance**.
- **Best Practice:** You are encouraged to **save early drafts ofd your PDE sketches and use Track Changes in MS Word** to demonstrate your writing process.

By remaining in this course, you agree to uphold these policies. If you anticipate any challenges regarding attendance, technology access, or other responsibilities, please communicate with the instructor early to seek guidance or accommodations.

Communications:

Clear and respectful communication is essential to your success in this course. Here's how you are expected to communicate with the instructor and how I will communicate with you:

How to Contact the Instructor

- **Email** is the primary method for reaching the instructor with official or personal questions. Please use your university email address and include the course name in the subject line.
- The instructor will normally respond to emails within **24 hours, Monday through Friday**.
- **Do not expect responses after 8:00 PM or on weekends** unless it is an emergency.
- If your question requires a more extended conversation, you may be asked to bring it to office hours.

- WeChat groups may be used informally for unofficial or emergency updates; however, please note that **WeChat is not a secure or university-approved platform** for professional or personal academic communication. While it may be convenient, it should **not** be used for submitting assignments, discussing grades, or sharing sensitive information.
- If **group messaging** is needed beyond university email or the Canvas course site, we will use **Microsoft Teams**, which is supported and approved by the university for official academic course communication.
- **Please note:** the instructor will not answer questions concerning grades or sensitive information over MS Teams. This type of communication will be conducted over university email or at office hours.

Office Hours

- Scheduled office hours are held:
- Office hours are the best time to discuss grades, assignments, course materials, research ideas, or any concerns.
- No appointment is necessary during office hours; however, you are welcome to email in advance if you'd like to reserve a specific time slot during office hours.
- If you are unable to meet during office hours, you can e-mail the instructor 24 hours in advance to schedule a time between**:** .

Course Site (Canvas)

- **All announcements, assignments, readings, rubrics, and grades** will be posted on **Canvas**.
- You are expected to **check both your university email and the Canvas course site daily** for updates, deadlines, and feedback.
- Do not rely solely on reminders—managing your time and staying informed is part of your responsibility in this course.

Discussion Guidelines:

Civility is an essential ingredient for academic discourse. All communications for this course should be conducted constructively, civilly, and respectfully. Differences in beliefs, opinions, and approaches are to be expected. Please bring any communications you believe to be in violation of this class policy to the attention of your instructor. Active interaction with peers and your instructor is essential to success in this course, paying particular attention to the following:

- Be respectful of others and their opinions, valuing diversity in backgrounds, abilities, and experiences.
- Challenging the ideas held by others is an integral aspect of critical thinking and the academic process. Please word your responses carefully, and recognize that others are expected to challenge your ideas. A positive atmosphere of healthy debate is encouraged.
- Read your online discussion posts carefully before submitting them.

Academic Integrity:

As a student, you should abide by the academic honesty standard of Duke Kunshan University. The DKU Community Standard states: "Duke Kunshan University is a community comprised of individuals from diverse cultures and backgrounds. We are dedicated to scholarship, leadership, and service and to the principles of honesty, fairness, respect, and accountability. Members of this community commit to reflecting upon and upholding these principles in all academic and non-academic endeavors, and to protecting and promoting a culture of integrity and trust." For all graded work, students should pledge that they have neither given nor received any unacknowledged aid.

Generative AI:

As outlined in the "What are the course policies?" section above.

Academic Policy & Procedures:

You are responsible for knowing and adhering to academic policy and procedures as published in the University Bulletin and Student Handbook. Please note, an incident of behavioral infraction or academic dishonesty (cheating on a test, plagiarizing, use of online tools prohibited by the instructor at the course or assignment level, etc.) will result in immediate action from me, in consultation with university administration (e.g., Dean or Associate Dean of Undergraduate Studies, Student Conduct, Academic Advising). Please visit the Undergraduate Studies website for additional guidance related to academic policy and procedures. Academic integrity is everyone's responsibility.

Academic Disruptive Behavior and Community Standard:

Please avoid all forms of disruptive behavior, including but not limited to: verbal or physical threats, repeated obscenities, unreasonable interference with class discussion, making/receiving personal phone calls, text messages or pages during class, excessive tardiness, leaving and entering class frequently without notice of illness or other extenuating circumstances, and persisting in disruptive personal conversations with other class members. Please turn off phones, pagers, etc. during class unless instructed otherwise. Laptop computers may be used for class activities allowed by the instructor during synchronous sessions. If you choose not to adhere to these standards, I will take action in consultation with university administration (e.g., Dean of Undergraduate Studies, Student Conduct, Academic Advising).

Academic Accommodations:

Duke Kunshan University makes reasonable academic accommodations for qualified students with disabilities. All undergraduate accommodations must be approved through the Student Accommodation Services. Students requesting accommodations for this course should forward their official accommodation letter to the instructor and ask to schedule a time to meet and discuss the implementation of their accommodation(s). It is the student's responsibility to meet, discuss, and provide an electronic copy of the Instructor Accommodation Letter to each instructor. Accommodations will not be granted retroactively. Accommodations for test, quiz, or exam taking must be arranged with the professor at least a week before the date of the quiz, test or exam, including finals.

What campus resources can help me during this course?

Academic Advising and Student Support

Please consult with me about appropriate course preparation and readiness strategies, as needed. Consult your academic advisors on course performance (i.e., poor grades) and academic decisions (e.g., course changes, incompletes, withdrawals) to ensure you stay on track with degree and graduation requirements. In addition to advisors, staff in the Academic Resource Center can provide recommendations on academic success strategies (e.g., tutoring, coaching, student learning preferences). Please visit the Office of Undergraduate Advising website for additional information related to academic advising and student support services.

Writing and Language Studio

For additional help with academic writing—and more generally with language learning—you are welcome to make an appointment with the Writing and Language Studio (WLS). You can register for an account, make an appointment, and learn more about WLS services, policies, and events on the WLS website.

IT Support

If you are experiencing technical difficulties, please contact IT:

- China-based faculty/staff/students 400-816-7100, (+86) 0512- 3665-7100
- US-based faculty/staff/students (+1) 919-660-1810
- International-based faculty/staff/students can use either telephone option (recommend using tools like Skype calling)
- Live Chat: https://oit.duke.edu/help & Email: service-desk@dukekunshan.edu.cn

What is the expected course schedule?

Week 1: Form = Process **Session 1:** Introduction to Creative Coding & Generative Form	
Date	2025/10/21
Lecture	Overview of creative coding and generative form. Students learn how artists use algorithms as compositional systems and explore the Processing IDE as a medium for rule-based design.
Texts	• Reas & McWilliams, *Form + Code in Design, Art, and Architecture* (Introduction & Ch. 1). https://archive.org/details/formcodeindesign0000reas • Aline Guillermet, "Vera Molnar's Computer Paintings." *Representations* 149 (2020): 1–30. https://www.jstor.org/stable/26908911 • Supplementary – Processing.org (Getting Started): https://processing.org
Discussion	What happens when an artist writes a process instead of drawing a form? Student facilitator connects Molnár's systems-based practice to Reas's idea of coded structure.
Studio	In-class Processing exercises on primitives, color, and interaction.
Assignment	• **Project 1:** Procedural Poster (1 image + README). • **Reflection (350–500 words):** Describe how writing a composition in code alters your approach to design. **Readings and assignments are due Saturday at midnight.**

Week 1: Form = Process **Session 2 Studio:** Form as Instruction	
Date	2025/10/23
Lecture	Short recap of form-as-process; live coding demonstration of random and parameterized shapes.
Texts	• Reas & McWilliams, *Form + Code in Design, Art, and Architecture* (Introduction & Ch. 1). https://archive.org/details/formcodeindesign0000reas • Aline Guillermet, "Vera Molnar's Computer Paintings." *Representations* 149 (2020): 1–30. https://www.jstor.org/stable/26908911 • Supplementary – Processing.org (Getting Started): https://processing.org
Discussion	Peer analysis of code snippets; facilitator highlights how variation emerges from rule modification.
Studio	Work session developing *Procedural Poster* sketches; instructor check-ins.
Assignment	Finalize Project 1 and upload .zip with .pde, exports, and reflection by deadline. **Readings and assignments are due Saturday at midnight.**

Week 2: Repetition & Rule **Session 1:** Rules, Chance & Authorship	
Date	2025/10/28
Lecture	Exploring instruction-based art and computational pattern-making.
Texts	• Sol LeWitt, "Paragraphs on Conceptual Art." *Artforum* (1967). https://robertspahr.com/teaching/studio2/readings/sol_lewitt.pdf

	• Nick Montfort et al., *10 PRINT CHR$(205.5 + RND(1)); : GOTO 10* (Introduction). https://10print.org/10_PRINT_121114.pdf • Frieder Nake, “There Should Be No Computer Art.” (1971). https://dam.org/museum/wp-content/uploads/2021/05/Nake1971-there-should-be-no-computer-art.pdf • Supplementary – LearningProcessing.com (Loops and Random).
Discussion	Facilitator leads comparison between LeWitt’s instructional logic and 10 PRINT’s pattern code. What is the relationship between rule and authorship?
Studio	Coding lab on loops, conditional statements, and random functions.
Assignment	• **Project 2:** Two Rule-Sets Pattern. • Create two generative rules that produce contrasting outputs. • Include a short reflection (400–600 words) on where you locate authorship within your rules. **Readings and assignments are due Saturday at midnight.**

Week 2: Repetition & Rule **Session 2 Studio:** Iteration & Variation	
Date	2025/10/30
Texts	• Sol LeWitt, “Paragraphs on Conceptual Art.” *Artforum* (1967). https://robertspahr.com/teaching/studio2/readings/sol_lewitt.pdf • Nick Montfort et al., *10 PRINT CHR$(205.5 + RND(1)); : GOTO 10* (Introduction). https://10print.org/10_PRINT_121114.pdf • Frieder Nake, “There Should Be No Computer Art.” (1971). https://dam.org/museum/wp-content/uploads/2021/05/Nake1971-there-should-be-no-computer-art.pdf • Supplementary – LearningProcessing.com (Loops and Random).
Studio	In-progress code review; peer feedback on generative variation. Instructor demos controlled randomness and rule switching.
Assignment	Finalize Project 2 and upload .zip with .pde, exports, and reflection by deadline. **Readings and assignments are due Saturday at midnight.**

Week 3: Transformation / Transcoding **Session 1:** Translating Images into Process	
Date	2025/11/04
Lecture	From bitmap to behavior — understanding transcoding and visual systems.
Texts	• John Maeda, *Design by Numbers* (Selections). https://archive.org/details/designbynumbers0000maed • Golan Levin, “Some Notes on Visualization Without Computers.” *Sonic Acts* (2006). https://www.flong.com/archive/texts/essays/essay_sonicacts_2006/index.html • Supplementary – Processing.org (PImage Reference).
Discussion	Facilitator leads on translation between data and image: what is lost or gained when we “write” an image in code?
Studio	Pixel access and image manipulation in Processing; pixel-sorting and mapping.

Assignment	• **Project 3:** Image Re-Written. • Transform an image algorithmically to reveal underlying structure. • Reflection (500–700 words) analyzing how code becomes a form of seeing. **Readings and assignments are due Saturday at midnight.**

Week 3: Transformation / Transcoding **Session 2 Studio:** Re-Coding the Image	
Date	2025/11/06
Texts	• John Maeda, *Design by Numbers* (Selections). https://archive.org/details/designbynumbers0000maed • Golan Levin, "Some Notes on Visualization Without Computers." *Sonic Acts* (2006). https://www.flong.com/archive/texts/essays/essay_sonicacts_2006/index.html • Supplementary – Processing.org (PImage Reference).
Studio	Code clinic on PImage functions, filter effects, and iteration. Peer critique of visual transformations.
Assignment	Finalize Project 3 and upload .zip with .pde, exports, and reflection by deadline. **Readings and assignments are due Saturday at midnight.**

Week 4: Parameterization & Systems **Session 1:** Complexity and Control	
Date	2025/11/11
Lecture	Exploration of parameterized systems and generative control structures.
Texts	• Philip Galanter, "What Is Generative Art? Complexity Theory as Context." (2003). https://www.philipgalanter.com/downloads/ga2003_paper.pdf • Margaret Boden & Ernest Edmonds, "What Is Generative Art?" *Digital Creativity* 20 (1–2) (2009). https://www.researchgate.net/profile/Ernest-Edmonds/publication/233128802_What_is_generative_art/links/00463521693ec679c8000000/What-is-generative-art.pdf • Supplementary – LearningProcessing.com (Functions & Objects).
Discussion	Facilitator: How do parameters shift creative control from artist to system?
Studio	Building function-based and class-based systems; exporting multiple variations.
Assignment	• **Project 4:** Composition System (x16). • Generate a set of 16 unique outputs from one system. • Reflection (400–600 words) on authorship and autonomy. **Readings and assignments are due Saturday at midnight.**

Week 4: Parameterization & Systems **Session 2 Studio:** Iterative Systems	
Date	2025/11/13
Texts	• Philip Galanter, "What Is Generative Art? Complexity Theory as Context." (2003). https://www.philipgalanter.com/downloads/ga2003_paper.pdf

	• Margaret Boden & Ernest Edmonds, “What Is Generative Art?” *Digital Creativity* 20 (1–2) (2009). https://www.researchgate.net/profile/Ernest-Edmonds/publication/233128802_What_is_generative_art/links/00463521693ec679c8000000/What-is-generative-art.pdf • Supplementary – LearningProcessing.com (Functions & Objects).
Studio	Mid-project critiques; parameter adjustment and debugging.
Assignment	Finalize Project 4 and upload .zip with .pde, exports, and reflection by deadline. **Readings and assignments are due Saturday at midnight.**

Week 5: Data as Material **Session 1:** Visualizing Meaning	
Date	2025/11/18
Lecture	Exploring data as aesthetic substance and ethical artifact.
Texts	• Harold Cohen, “The Further Exploits of AARON, Painter.” (1995). https://www.kurzweilcyberart.com/aaron/pdf/furtherexploits.pdf • Supplementary – TheNatureOfCode.com (Random & Distributions): https://natureofcode.com • Processing.org (LoadTable Reference).
Discussion	Facilitator leads debate: When does visualizing data become interpretation or storytelling?
Studio	Mapping data to form and motion; creating a “poetic” data visualization.
Assignment	• **Project 5:** Poetic Data Graphic. • Visualize a small dataset creatively. • Reflection (400–600 words) on how data takes on aesthetic meaning. **Readings and assignments are due Saturday at midnight.**

Week 5: Data as Material **Session 2 Studio:** Data-Driven Design	
Date	2025/11/20
Texts	• Harold Cohen, “The Further Exploits of AARON, Painter.” (1995). https://www.kurzweilcyberart.com/aaron/pdf/furtherexploits.pdf • Supplementary – TheNatureOfCode.com (Random & Distributions): https://natureofcode.com • Processing.org (LoadTable Reference).
Studio	Troubleshooting data loading and mapping; peer reviews.
Assignment	Finalize Project 5 and upload .zip with .pde, exports, and reflection by deadline. **Readings and assignments are due Saturday at midnight.**

Week 6: Simulation & Emergence **Session 1:** Systems That Behave	
Date	2025/11/25

Lecture	Agents, rules, and emergence in digital art.
Texts	• Revisit Boden & Edmonds (2009) sections on creativity and system behavior. • Supplementary – TheNatureOfCode.com (Autonomous Agents & Flocking).
Discussion	Facilitator asks: What aesthetics emerge when systems act independently?
Studio	Agent and particle simulation coding lab.
Assignment	• **Project 6: Tiny Ecosystem.** • Design a simulation with two agent types and document emergent interactions. • Reflection (500–700 words). **Readings and assignments are due Saturday at midnight.**

Week 6: Simulation & Emergence **Session 2 Studio:** Emergent Behaviors	
Date	2025/11/27
Texts	• Revisit Boden & Edmonds (2009) sections on creativity and system behavior. • Supplementary – TheNatureOfCode.com (Autonomous Agents & Flocking).
Studio	Work session and critique on simulation behaviors; capture short video export.
Assignment	Finalize Project 6 and upload .zip with .pde, exports, and reflection by deadline. **Readings and assignments are due Saturday at midnight.**

Week 7: Portfolio & Reflection **Session 1:** Synthesis & Curation	
Date	2025/12/02
Lecture	Review and synthesis of major themes: form, rule, data, behavior.
Texts	• Lauren McCarthy, project statements (*Follower*, *Social Turkers*). https://lauren-mccarthy.com/ • Kate Compton, “Casual Creators.” *ICCC 2015 Proceedings*. https://computationalcreativity.net/iccc2015/proceedings/10_2Compton.pdf • Supplementary – Processing.org (Export Guidelines).
Discussion	Facilitator leads closing conversation on authorship and collaboration in code-based art.
Studio	Portfolio organization; essay draft peer review.
Assignment	Begin final essay (1,000–1,200 words) and curate six projects for portfolio. **Readings and assignments are due Saturday at midnight.**

Week 7: Portfolio & Reflection **Session 2 Studio:** Final Critique & Submission	
Date	2025/12/04
Studio	Final presentations of portfolios; peer and instructor critiques.
Assignment	Submit **Final Portfolio + Reflective Essay** via Canvas (see submission guidelines). **Final Portfolio + Reflective Essay are due by midnight.**