Admin Brief

Programming by Design: Computing, Representation, and Reasoning

Grade 9 Computer Science — Instructor: Eric Allatta

Course Summary

This full-year course introduces computer science as a discipline of structured reasoning, communication, and system design. Students explore computing as both a technical and cultural practice: one grounded in logic and abstraction, but inseparable from authorship, ethics, and representation.

Rather than start with syntax, students begin by investigating how structured thinking allows us to build systems, interpret information, and design solutions. They move from models and metaphors to code and communication—building fluency across representations and reflecting on what it means to "think computationally."

Design Priorities

- **Structure Before Syntax:** Students internalize precision and logic through systems thinking before writing code.
- **Code as Communication:** Code is treated as expressive language—designed for both humans and machines.
- **Fluency Across Representations:** Students translate between diagrams, markup, functions, data, and procedural steps.
- **Equity + Authorship as Threads:** Cultural analysis, digital authorship, and bias are embedded throughout, not siloed.
- **Routines Over Gimmicks:** Repeated practices (journaling, interpretation, revision) structure long-term fluency.

Strategic Outcomes

- **Foundational Literacy for Advanced Study:** Students finish with conceptual readiness for AP CS A, data science, and web systems.
- **Publicly Presentable Work:** Blogs, systems diagrams, journals, and code artifacts that reflect real learning and reasoning.
- **Cross-Disciplinary Thinking:** Students develop functional reasoning and abstraction aligned to Algebra 2, history, and civics.

Key Instructional Features

- **Unit 0 — The Story of Data:** A narrative, non-coding introduction that seeds habits of reflection, structure, and representation.
- **Threaded Routines:** Weekly journals, prompt logs, blog posts, and system diagrams that spiral across units.

- **Assessment as Reflection:** Student work is assessed for clarity, structure, and revision—not correctness alone.
- **Real Tools in Context:** Students use CLI, markdown, version control, Pyret, Python, APIs, and HTML/CSS meaningfully—not artificially.

Implementation Notes

- Modular design supports differentiated pacing, collaborative planning, and interdisciplinary extensions.
- Materials are aligned to Bootstrap, Math for America, and NYC DOE CS pathways.
- Course is built for real classrooms: 180-day pacing, reflection-driven grading, and scaffolded supports for all learners.

Contact

For questions, materials, or full syllabus access, contact:
Eric Allatta — Curriculum Lead
Email: ericallatta@gmail.com