Technological advances, particularly in computing, are a driving force in contemporary architectural design. Mastering the state-of-the-art requires more than knowledge of software commands; it requires computational thinking. Designers, accustomed to drawing each line and surface on their screen, must learn to work computationally, through the specification of processes, rules, and relationships.

This course introduces students to the fundamental concepts of computation through explorations with generative scripting and parametric tools. The goal is to understand the potential of computation and the role it can play as part of one’s design process; not as a collection of specific tools, but as a way of thinking about design. By the end of the semester, students will have a vocabulary and an understanding of computing that will inform their future explorations with more advanced tools and technologies.

Objectives

- Acquire essential computational design skills
- Acquire knowledge of fundamental computational concepts, their history, and how they relate to design and architecture
- Acquire knowledge of important figures and projects within and related to computational design
- Acquire a critical understanding of the strengths and limitations of various kinds of computing
- Acquire strategies for computing efficiently and effectively

Method

This course is uses an active learning format with a weekly lab session followed by a lecture. Skills homework is delivered online and provides students with the opportunity to learn and practice new tools and techniques. Lab sessions demonstrate more complex concepts and methods in an interactive setting. Following the labs, online lectures set up peer discussions in the Kennedy active learning space which interrogate course topics and frame them within an architectural and historical context. Short design projects invite students to examine the analytic and expressive potential of computational methods. A final project asks students to critically reflect on the ideas of the course and how these relate to their developing understanding of design.

Software Proficiency

A basic knowledge of computers and experience with 3D modeling, particularly Rhinoceros, is highly recommended. No prior programming experience is expected or required.