PREMISE
From algorithmic design to open source and from virtual reality to robotics, many designers are seeking new tools to explore formal and functional properties as sources of ordering systems. This is an introductory course which focuses on computational techniques in architecture as they specifically relate to design processes and procedures. The topics of the course will borrow concepts from computer science, computational geometry and other fields and adapt them to specific design problems such as design development, fabrication, robotics, material simulation, and environmental analysis. A specific area of interest for the course will investigate the relationship between computational methods and fabrication processes. Using the past as a frame of reference for what is to come, we will examine the history of machine epistemology in architecture and draw from a variety of precedents ranging from early 19th century and to the present and investigate how design actualization relies on available tools and knowledge of techniques.

OBJECTIVE
The objective of the course is to help students become better informed users of digital tools and develop the skills necessary for creating or manipulating computational solutions for specific design problems, which include geometry generation and manipulation, analysis of data and design evaluation. As part of the course, students will acquire some hands on experience in parametric modeling and programming as this is the craft that underpins computational design.

METHOD
Class will meet on twice a week. There will be weekly assignments to provide students with the opportunity to learn and practice new tools and techniques. The lab sessions allow for the demonstration of concepts and methods in an interactive setting. Fridays will be given over to lecture and discussion. The lectures will frame larger topics within an architectural and historical context while also providing time for further instruction on complex topics.

The mid-term project asks students to analyze a precedent project and develop as set of diagrams with a parametric model. The final project invites students to collaboratively engage in computational thinking through analogue and digital methods and requires a set of compositions (i.e., drawings, paintings, carvings) produced with the robotic arm.

SOFTWARE
All required software is available on SoA lab computers, but students will find that acquiring student versions of the required software is necessary to execute their coursework. Rhinoceros (Rhino) is required for the course. Grasshopper (grasshopper3d.com) is a free plugin for Rhinoceros that you will also need. Grasshopper runs on a Windows operating system. Be aware that Parallels runs incredibly slow and will prohibit the efficiency of work production. Be sure to have it installed on your machine in preparation for class. Please be prepared to bring your laptop to each class.

GRADING
Final course grades are based upon the following rubric:
- 10% Attendance and Participation
- 40% Weekly Assignments
- 25% Midterm Project
- 25% Final Project

PREREQUISITES
ARCH 2101/2102 for undergraduates / ARCH_6101/6102 or M2 standing for graduates.
A basic knowledge of computers and experience with 3D modeling, particularly Rhinoceros, is highly-recommended. No prior programming experience is expected or required.