Course Description
The art and science of computation predates the modern computer by centuries, and in fact is built upon a double foundation of formal logic (going all the way back to authors such as Aristotle) and mathematical reasoning such as set theory, algorithms, and functions. The modern computer is a physical model for repeated uses Boolean logic in the same way that the abacus is a physical model for repeated steps for performing numerical computation.

The goals of this course are to give students (1) a grounding in formal logic from Aristotle through Boole, (2) a foundation in the mathematics relevant to the theory of computation, (3) the philosophical connection between the two, (4) initiation into seeing the world through the lens of computation, which is proving a rethinking of the world on a scale at least as large as the introduction of calculus caused, and (5) an understanding of the limits of formal logic and computation, including the need for heuristics alongside algorithms because of either computational complexity or computational impossibility. All of this is oriented towards the end of getting students to think computationally and algorithmically.

Twenty-five out of a total of 36 weeks of the course are spent on pure logic and the Theory of Computation. In addition to these 25 weeks, the final project is a five-week project at the end of the course. This project also has significant mathematical content, but the specifics changes somewhat form year to year based on the project itself. See below for the first-year project and mathematical topics involved.

In the six weeks that are not specifically mathematical or dedicated to the final project, the mathematical content is reinforced by utilizing programing concepts, principally using Java. However, the use of the Java programing language is to help establish these mathematical principles. For this reason, this course should be considered a mathematics course and not a technology course.
Particular Mathematical Topics

1. Formal logic
   a. Distinction between formal and material logic
   b. Terms, propositions, truth, and validity
   c. Compound propositions (“and” versus “or”)
   d. Hypothetical propositions (if-then) and biconditionals (if and only if)
   e. Truth Tables (proof of logical equivalence)
      i. Commutativity
      ii. Associativity
      iii. Double negation
      iv. DeMorgan’s Laws
      v. Set intersection, union, complement, and subset (with proofs) that support the logical operations.
   f. Categorical Propositions
   g. Quantificational propositions (“for all”, “there exists”)
   h. Inverse, Converse, and Contrapositive
      i. Truth table proofs
      ii. Relationship with *modens ponens* and *modus tollens*
   i. Syllogistic reasoning and the connection to quantificational propositions and hypothetical (if-then) propositions.
      i. Proof techniques related to syllogistic reasoning.
      ii. Conjunctive and Disjunctive syllogisms
      iii. Fallacies related to logic reasoning (and their appearance in errant mathematical proofs)

2. Number Theory
   a. Modular arithmetic
   b. Numbers systems in bases other than 10 (binary, octal, hexadecimal)
      i. Representation
      ii. Conversion (decimal to binary, binary to hexadecimal, octal to decimal, etc.)
      iii. Algorithms (addition, subtraction, multiplication, division)
      iv. Rounding errors for finite representation of numbers in various bases (binary numbers vs. decimal numbers)
   c. Number theoretic algorithms and concepts
      i. GCD (Euclid)
      ii. Sieve of Eratosthenes
      iii. Other methods of testing primality
      iv. Factoring algorithms
      v. Fibonacci numbers
      vi. Collatz conjecture
      vii. Twin prime conjecture
      viii. LCM (as related to GCD)
      ix. Summation formulas
Summary of Mathematical Topics

Logic and Computation

3. Theory of Computation
   a. Logic Gates
   b. Disjunctive Normal Form
   c. Formal languages and regular expressions
   d. Deterministic finite automata
      i. Non-deterministic finite automata
e. Kleene’s Theorem
f. Complexity Analysis
   i. $O(n)$ analysis
   ii. Limits and asymptotic behavior for complexity functions
      1. (Note that knowledge of limits from Calculus is required as it is used to define measures of complexity and to evaluate the equivalence classes for functions using certain methods.)
      2. (Note that heavy use of logarithmic, exponential functions, polynomial functions, and combinations of these types of functions using all operations, including composition, are all required here.)
   iii. The classes of P, NP, and NP-Complete algorithms
g. Turing machines
   i. Definition of algorithm as a Turing machine (this involves using the formal definition of a function, etc.)
   ii. Design of Turing Machines to accomplish number theoretic tasks (conversion to binary, etc.)
   iii. Power of the universal Turing Machine
      1. The Church-Turing Thesis
   iv. The Halting Problem
   v. Other undecidable problems
      1. Hilbert’s 10th problem (integer solutions to multi-variable polynomials)
      2. Improper Integration (convergence and divergence)
      3. Post’s correspondence Problem
   vi. Severe Limits on Computation
      1. Hilbert’s third problem (*Entscheidungsproblem*)
      2. Gödel’s incompleteness theorem

4. Miscellaneous Topics
   a. Formal definition of a function for domains other than numbers.
   b. Matrices and arrays (vectors)
      i. Indexing and operations (add, multiply, scalar multiplication)
   c. Recursive functions
      i. Recursive versus explicit formulas
      ii. Summation formulas (sums of consecutive integers, sums of squares, sums of cubes) used to derive explicit formulas from recursive formulas.
Summary of Mathematical Topics

Logic and Computation

- **d. Hash functions and randomness**
- **e. Uncertainty and Heuristics**
  - i. Using statistical techniques for decision making on incomplete data and information
    1. Modeling physical processes, constructing hypotheses, estimating the quality of a data source, quantifying uncertainty, identifying patterns in data.
    2. Time-varying averages
  - ii. Heuristic functions
- **5. Final Project Topics [Particular topic may change from year to year]**
  - a. RSA Encryption
  - b. Fermat’s Little Theorem (as a corollary to Euler’s Theorem)
  - c. The Chinese Remainder Theorem