

1. **Department, number and title of course:** Computer Science: CS 70, Discrete Mathematics and Probability Theory
2. **Catalog Description:** (4 units) Logic, infinity, and induction; applications include undecidability and stable marriage problem. Modular arithmetic and GCDs; applications include primality testing and cryptography. Polynomials; examples include error correcting codes and interpolation. Probability including sample spaces, independence, random variables, law of large numbers; examples include load balancing, existence arguments, Bayesian inference.
3. **Prerequisites:** CS 61A, Math 1A, Math 1B (or equivalents).
4. **Textbooks and/or other required material:**
 - Kenneth H. Rosen, *Discrete Mathematics and its Applications, 4th Edition*, McGraw-Hill, Inc., New York, 1995.
 - Michael T. Goodrich and Roberto Tamassia, *Data Structures and Algorithms in Java*, John Wiley & Sons, 1998.
5. **Course objectives:** The goal of this course is to introduce students to ideas and techniques from discrete mathematics that are widely used in Computer Science. The course aims to present these ideas "in action"; each one will be geared towards a specific significant application. Thus, students will see the purpose of the techniques at the same time as learning about them.
6. **Topics covered:**
 - Propositions and Proofs
 - Mathematical Induction: recursion, the stable marriage problem
 - Propositional Logic: automated proof and problem-solving
 - Arithmetic Algorithms: gcd, primality testing, the RSA cryptosystem
 - Polynomials and their Applications: error-correcting codes, secret sharing
 - Probability and Probabilistic Algorithms: load balancing, probabilistic constructions, conditional probability, Bayesian inference
 - Diagonalization, Self-Reference and Uncomputability
7. **Class/laboratory schedule:** Three hours of lecture per week, or three hours of lecture and two hours of discussion per week.
8. **Contribution of course meeting the professional component:** This course contributes to mathematics topics.
9. **Relationship of course to program objectives:** This course requires students to apply their fundamental knowledge of mathematics, science, and engineering to analyze and solve computer engineering problems. The students learn to identify, formulate and solve challenging problems. They learn to design a system, component or process to meet desired needs. The course also provides practice in communicating mathematical content clearly.
10. **Updated by:** David Wagner; March 27, 2006.