CS 135 Spring 2009
Syllabus

Stephen L. Bloom

The aim of the course is to integrate knowledge of basic mathematics with the problems involving specification, design and computation. By the end of the course, the student should be able to

- use sets, functions, lists, relations in the specification and design of problems
- use properties of arithmetic, modular arithmetic (sum, product, exponentiation), prime numbers, greatest common divisor; factoring, Fermat's little theorem
- make use of binary, decimal and base b notation systems, and translation methods
- use induction to design and verify recursive programs

and how to implement in Scheme all algorithms considered during the course.

Weekly topics

1. Introduction to Scheme; lists and functions.
2. More on Scheme; Induction; list induction.
3. Sets as Lists; project: implementing set operations of union, intersection, cartesian product, ... in Scheme.
4. Injective, surjective and bijective functions; Scheme programs for function composition, testing for injectivity, etc.
5. number notation systems; translating between base b and base c systems; implementation in Scheme.
6. Representing functions in various ways: lists of pairs, as a number (via base n notation); enumerating all functions $[n] \to [p]$.

7. Pigeon hole principle; orbits of functions; Scheme program to compute orbits
8. The boolean algebra of relations
9. partial, and total orders; lexicographic orders
10. some relational data base applications
11. equivalence relations, hash tables
12. The Euclidean algorithm and its implementation in Scheme.
13. Cardinality, Cantor's diagonal argument, uncountable sets
14. Graphs and their implementations
15. Connected components; Shortest paths.

Possible texts

- *The Little Schemer*, Friedman and Felleisen, MIT Press, and
- *Discrete Structures, Logic and Computability*, Hein, Jones and Bartlett Computer Science.