Math 180 – Fundamentals of Discrete Mathematics

Course Description from Bulletin: Basic counting techniques, discrete probability, graph theory, algorithm complexity, logic and proofs, and other fundamental discrete topics. (3-0-3)

Enrollment: Required for ITM majors. This course does not count toward any mathematics requirements in computer science, engineering, mathematics, or natural science degree programs.

Textbook(s): Discrete Mathematics with Applications, 4th edition, Susanna S. Epp.

Other required material: None

Prerequisites: None

Objectives:

1. Students will be able to express mathematics and other concepts in terms of formal logic and vice-versa, and usefully manipulate propositional and predicate logic.
2. Students will be able to evaluate and write simple mathematical proofs.
3. Students will compute time complexity of algorithms and understand time complexity from a mathematical viewpoint.
4. Students will know a certain amount about common concepts in discrete mathematics including graph theory, combinatorics, sets, relations, probability, binary arithmetic, recursion, and the pigeonhole principle.

Lecture schedule: 2 75 minute (or 3 50 minute) lectures per week

Course Outline: Hours

1. Fundamental Concepts – sets, Cartesian products, relations and databases, general pigeonhole principle, binary, octal, hexadecimal arithmetic, summation notation, recurrence relations, with applications. If time, one-to-one and onto functions. **9**
2. Logic and proof – expressing real-life concepts and mathematics in logic and vice-versa. Propositions, predicates, quantifiers, implication and equivalence. Basic proof techniques including the contrapositive, counterexamples, contradiction, and when to use each. **12**
3. Counting techniques and discrete probability – sum and product, permutations and combinations, binomial coefficients, alongside probability and expectation **8**
4. Graphs and rooted trees – degrees, paths, cycles, connectivity, isomorphism classes, applications **4**
5. Algorithm efficiency – algorithms (pseudocode), order of a function, big-Oh notation, complexity: worst-case and average-case, applications including recursion, search, and sort. **6**

Total: **39** hrs

Assessment:	Homework	10-30%
	Quizzes	0-20%
	Mid-Term Exams	20-60%
	Project	0-20%
	Final Exam	30-40%

Syllabus prepared by: Michael Pelsmajer, Rob Ellis, and Ben Reiniger

Date: 6/22/2017