

**CS 330 — Discrete Structures**  
**Syllabus (version 1.2)**  
**Final except for correcting typos or unforeseeable**  
**circumstance**

**Gruia Calinescu**  
**Department of Computer Science**  
**Illinois Institute of Technology**

**Spring 2012**

## **1 Course Summary**

Welcome to CS 330 Discrete Structures. This class has two major thrusts which complement each other nicely: increasing the mathematical sophistication (proofs and logic) and introducing the mathematical foundation of computer science. The emphasis is on understanding the material rather than simply accumulating information.

The class consists of two weekly (standard) lectures and one recitation session (officially called “lab”, Fridays 11:25-12:15, normally administered by the TA) in which students solve problems related to the lectures.

The instructor is going to a conference March 25-30. Thus there will be one week with recitations only, while Jan. 20, Feb. 3, and March 2 normal classes will be held instead of recitations.

## **2 Textbook**

The required textbooks is *Discrete Mathematics and Its Applications* by Kenneth H. Rosen, WCB/McGraw Hill, 7th edition, ISBN-10: 0073383090, ISBN-13: 978-0073383095. The sixth edition would be O.K. for learning, but may not be for assigned homework. Recommended are *Student’s Solution Guide to Discrete Mathematics and Its Applications* by Kenneth H. Rosen, WCB/McGraw Hill, 6th Edition, ISBN: 0073107794, and *Schaum’s Outline of Discrete Mathematics, 3rd Ed. (Schaum’s Outlines)* (Paperback) McGraw-Hill; 3rd edition ISBN 0071470387.

These books should be available at the bookstore and on the web. Our library has the textbook on reserve.

### 3 Current Catalog Description

Introduction to the use of formal mathematical structures to represent problems and computational processes. Topics covered include Boolean algebra, first-order logic, recursive structures, graphs, and abstract language models. Co-requisite: CS 116 or CS 201. (3-0-3)

### 4 Course Goals

Students should be able to:

1. Illustrate by examples the basic terminology of functions, relations, and sets and demonstrate knowledge of their associated operations.
2. Demonstrate in practical applications the use of basic counting principles of permutations, combinations, inclusion/exclusion principle and the pigeonhole methodology.
3. Calculate probabilities of events and expectations of random variables for problems arising from games of chance
4. Establish and solve recurrence relations that arise in counting problems including the problem of determining the time complexity of recursively defined algorithms.
5. Model logic statements arising in algorithm correctness and real-life situations and manipulate them using the formal methods of propositional and predicate logic.
6. Outline basic proofs for theorems using the techniques of - direct proofs, proof by counterexample, proof by contraposition, proof by contradiction, mathematical induction.
7. Relate the ideas of mathematical induction to recursion and recursively defined structures.
8. Illustrate by example basic terminology of graph theory and model problems in computer science using graphs and trees.
9. Deduce properties that establish particular graphs as Trees, Planar, Eulerian, and Hamiltonian.
10. Illustrate the application of trees and graphs to data structures.
11. Explain the basic concepts modeling computation including formal machines, languages, finite automata, Turing machines

## 5 Getting Help

Reasonable accommodations will be made for students with documented disabilities. In order to receive accommodations, students must obtain a letter of accommodation from the Center for Disability Resources and make an appointment to speak with me as soon as possible. The Center for Disability Resources is located in the Life Sciences Building, room 218, 312-567-5744 or [disabilities@iit.edu](mailto:disabilities@iit.edu).

Office hours are Monday 3:15-4:15 and Wednesday 1-2, in room SB 229B, or by appointment. For an appointment send e-mail to [calinescu@iit.edu](mailto:calinescu@iit.edu) or call 312-567-5273. Please spend a little time trying to understand yourself a homework problem before asking for help.

Some handouts (including this syllabus, homeworks, and some solutions) will be available at <http://www.cs.iit.edu/~cs330>.

The TA for the class is Francis C. Lockom ([flockom@hawk.iit.edu](mailto:flockom@hawk.iit.edu)) and has office hours Tuesday 1-2 PM and Friday, from the end of the class until 2PM, in room SB 013, 419-509-4997. Students are expected to check email every week day of the semester. Clarifications on assignments or other important announcements might be sent by email - and will also be posted on the web page.

## 6 Grading

The grading allocation is given below.

Homeworks	40%
Midterm 1	10%
Midterm 2	10%
Final exam	20%
Quizzes	10%
Class Participation	10%

The first midterm will be on Feb. 17, instead of recitation. The second midterm will also be instead of recitation, on March 16. The final exam will be held as scheduled by the University, on Tuesday, May 1, 8:00 am to 10:00 am. Twelve homeworks will be assigned (due every Wednesday other than exam weeks). You can discuss the problems with each other, but must write the solutions individually. Seek help from me if you are having difficulties with the homework. Except for extraordinary circumstances, homeworks will be accepted at latest one week late. The penalty for late submission is 5%, for two days late, 10% for five days late, and 20% for one week late.

The first midterm will be open-books and closed notes. Precisely, each student can bring either the sixth or seventh edition of the textbook, printed version only, with no handwritten notes on the book. Electronic versions will not be allowed; however a printout from the textbook

(only) of maximum 50 double-sided pages may be used provided it is given to the instructor the lecture before the exam. If the midterm is “successful”, then the second midterm will also allow the textbook, and possibly the final as well. Otherwise, the second midterm and the final exam are closed books and closed notes, except for some notes distributed by me (photocopied key terms and results). The exams will be strictly individual, and will contain (slightly modified) homework problems. Standard departmental policy regarding academic (dis)honesty applies. There will be no make-up exams, other than the final exam. Exams missed for valid reasons will be replaced by a higher weight of the final.

There will be 13 quizzes (15-30 minutes each), each Monday other than weeks with exams. Quiet discussion within small groups during the quizzes is allowed. A “small group” consists of up to three students, sitting in the same row with one of them at the end of the row. The lowest three quiz scores will be dropped when computing the composite score.

Attendance (including the recitation) is mandatory and will be reflected in the “class participation” score (valid excuses are required after the third missed class/recitation). Also, disturbing the class will decrease this score. Moreover, students will be assigned reading from the textbook to competently discuss in class. Exceptions from the attendance: students whose score reflect an A may go to recitations only (no score is not A).

Every electronic device (anything with an on/off button) should be off during the class (exception: disability-helping devices).

Each score will be scaled by the highest in the class. Then a composite score will be computed, and grades assigned almost based on the ranges: A=90-100 B=80-89 C=70-79 D=60-69 E=0-59. There is no prescribed curve and you are not competing with each other.

## 7 Topics to be covered

- Sets, Functions and relations - sets, set operations, functions, summations, growth of functions, equivalence relations, countable and uncountable sets, examples of algorithm analysis
- Counting Methods permutations, combinations, discrete probability, pigeonhole principle
- Advanced counting inclusion-exclusion, recurrence relations, methods of solving recurrences, examples from computer sciences
- Introductory Logic propositional logic, predicate logic, proof methodologies, examples of algorithm correctness
- Introduction to Graphs - trees , connectivity, Eulerian traversals, minimum spanning tree, planarity, Euler formula, matchings
- Languages