CS 250 - Discrete Structures I, Portland Community College

12000 SW 49th Ave, Portland, OR 97219
Doug Jones
cdjones@pcc.edu

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
Discrete Structures (DS)	26
Algorithms and Complexity (AL)	4

Where does the course fit in your curriculum?

CS 250 is the first course in a two-term required sequence in discrete mathematics for Computer Science transfer students. Students typically complete the sequence in their second year.

College algebra and 1 term of programming are pre-requisites for CS 250. The second course in the sequence (CS 251) requires CS 250 as a pre-requisite.

Approximately 80 students per year complete the discrete mathematics sequence (CS 250 and CS 251).

What is covered in the course?

- Introduction to the Peano Axioms and construction of the natural numbers, integer numbers, rational numbers, and real numbers.
- Construction and basic properties of monoids, groups, rings, fields, and vector spaces.
- Introduction to transfinite ordinals and transfinite cardinals, and Cantor's diagonalization methods
- Representation of large finite natural numbers using Knuth's "arrow notation"
- Introduction to first order propositional logic, logical equivalence, valid and invalid arguments
- Introduction to digital circuits
- Introduction to first order monadic predicate logic, universal and existential quantification, and predicate arguments
- Elementary number theory, prime factors, Euclid's algorithm
- Finite arithmetic, Galois Fields, and RSA encryption
- Proof techniques, including direct and indirect proofs, proving universal statements, proving existential statements, proof forms, common errors in proofs
- Sequences, definite and indefinite series, recursive sequences and series
- Developing and validating closed-form solutions for series
- Well ordering and mathematical induction
- Introduction to proving algorithm correctness
- Second order linear homogeneous recurrence relations with constant coefficients
- General recursive definitions and structural induction
- Introduction to classical (Cantor) set theory, Russell's Paradox, introduction to axiomatic set theory (Zermelo-Fraenkel with Axiom of Choice).
- Set-theoretic proofs
- Boolean algebras
- Halting Problem

What is the format of the course?

CS 250 is a 4 credit course with 30 lecture hours and 30 lab hours. Classes typically meet twice per week for lecture, with lab sessions completed in tutoring labs outside of lecture.

Course material is available online, but this is not a distance learning class and attendance at lectures is required.

How are students assessed?

Students are assessed using in-class exams and homework. There are 5 in-class exams that count for 40% of the student's course grade, and 5 homework assignments that account for 60% of the student's course grade. In-class exams are individual work only, while group work is permitted on the homework assignments.

It is expected that students will spend 10 to 15 hours per week outside of class time completing their homework assignments. Surveys indicate a great deal of variability in this - some students report spending 6 hours per week to complete assignments, other report 20 or more hours per week.

Course textbooks and materials

The core text is *Discrete Mathematics with Applications* by Susanna S. Epp (Brooks-Cole/Cengage Learning). The text is supplemented with instructor-developed material to address topics not covered in the core text.

Students are encouraged to use computer programs to assist in routine calculations. Many students write their own programs, some use products such as Maple or Mathematica. Most calculators are unable to perform the calculations needed for this course. No specific tools are required.

Why do you teach the course this way?

This is a transfer course designed to meet the lower-division requirements of Computer Science and Engineering transfer programs in the Oregon University System with respect to discrete mathematics. As such, it serves many masters - there is no consistent set of requirements across all OSU institutions.

The majority of Portland Community College (PCC) transfer students matriculate to Portland State University, Oregon Institute of Technology, or Oregon State University, and these institutions have the greatest influence on this course. PCC changes the course content as needed to maintain compatibility with these institutions.

The most recent major course revision occurred approximately 24 months ago, although minor changes tend to occur every Fall term. Portland State University is reviewing all of their lower-division Computer Science offerings, and when they complete their process PCC expects a major revision of CS 250 and CS 251 will be required.

Students generally consider the discrete mathematics sequence to be difficult. Most students have studied some real number algebra, analysis, and calculus, but often have very limited exposure to discrete mathematics prior to this sequence.

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours
AL	Basic Analysis	Differences among best, expected, and worst case behaviors Big-O, Big-Omega, Big-Theta definitions Complexity classes Note: Remainder of Basic Analysis topics covered in CS 251	4
DS	Basic Logic	Propositional logic, connectives, truth tables, normal forms, validity, inference, predicate logical, quantification, limitations	10

DS	Proof Techniques	Implications, equivalences, converse, inverse, contrapositive, negation, contradiction, structure, direct proofs, disproofs, natural number induction, structural induction, weak/string induction, recursion, well orderings	10
DS	Basics of Counting	Basic modular arithmetic Other counting topics in CS 251	2
DS	Sets, Relations, Functions	Sets only: Venn diagrams, union, intersection, complement, product, power sets, cardinality, proof techniques. Relations and functions covered in CS 261	4

Additional topics

Elementary number theory, Peano Axioms, Zermelo-Fraenkel Axioms, Knuth arrow notation, simple digital circuits, simple encryption/decryption