Overview
COP 6556 is a graduate-level course on formal semantics of programming languages. Students will learn the fundamental concepts and approaches in defining the formal semantics of programming languages. These formal approaches lay the foundation for understanding, designing, and implementing new programming languages, and for ensuring program correctness.

Prerequisites
Students need to know discrete mathematics such as set, functions, and logic. Knowledge of abstract computational models (covered in COT 5420) will be very helpful. Knowledge of some high-level programming languages is also useful.

Text

Homework
There will be 5 homework assignments. Homework will be assigned every 2 weeks. Each homework assignment should be turned in at the end of class on the due date. Late homework turned in before the next lecture will receive partial credit. Homework assignments need to be typed.

Exams
There are two exams on Feb. 17 and April 22, respectively.

Grading
Homework (50%) + Exams (50%)

Tentative Schedule
Week 1 – Basic Mathematical Concepts: Logic, Sets, Functions, Relations, Partially Ordered Sets
Week 2 – Semantics of Sequential Programs – Operational Semantics
Week 3 – Principles of Induction and Inductive Definitions
Week 4 – Semantics of Sequential Programs – Denotational Semantics
Week 5 – Semantics of Sequential Programs – Axiomatic Semantics
Week 6 – Completeness of the Hoare Rules
Week 7 – Introduction to Domain Theory
Week 8 – Recursion Equations
Week 9 – Techniques for Recursion
Week 10 & 11 – Languages with Higher Types
Week 12 & 13 – Recursive Types
Week 14 & 15 – Nondeterminism and Parallelism