COP 6556 – Semantics of Programming Languages

Catalog Description
This course provides an overview of systematic and effective approaches to programming abstraction; formal specification techniques; program verification and; semantics of programming languages. (3 credits)

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.

Type
Can be an elective for MSCS, and Ph.D.

Course Objectives
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.

Topics
Basic Mathematical Concepts: Logic, Sets, Functions, Relations, Partially Ordered Sets
Semantics of Sequential Programs – Operational Semantics
Principles of Induction and Inductive Definitions
Semantics of Sequential Programs – Denotational Semantics
Semantics of Sequential Programs – Axiomatic Semantics
Completeness of the Hoare Rules
Introduction to Domain Theory
Recursion Equations
Techniques for Recursion
Languages with Higher Types
Recursive Types
Nondeterminism and Parallelism

Textbook
Glynn Winskel, *The Formal Semantics of Programming Languages – An Introduction*, The MIT Press 1993.

References
Eike Best, *Semantics of Sequential and Parallel Programs*, Prentice-Hall International, 1996.

Last Update
Xudong He 9/8/2012