Applied Logic	**Lecture 1 - Aug. 23**
CS 4860 Fall 2012	**Robert L. Constable**

Professor Robert Constable
Professor Christoph Kreitz
Abhishek Anan, TA

1 Course work

In addition to the required reading of *First-Order Logic* by R. Smullyan and the lecture notes by Constable & Kreitz, the course home page lists other recommended reading.

6 or 7 problem sets (every other week)	$\approx$ 40 - 50 %
course project	$\approx$ 15%
in class prelim (Oct 11)	$\approx$ 10%
final exam	$\approx$ 25-30%
4 or 5 in class problem solving sessions	$\approx$ 10%

2 Framework of the course

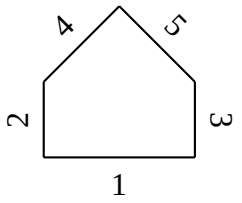
(1) Logic is the oldest academic discipline, grounded in philosophy and mathematics.

(2) Logic provides the "rules of evidence" for science and rational discourse (e.g. non-political fact base investigations).

(3) Logic provide the deductive and semantic *foundations* for mathematics, computer science and other theory based studies.

(4) The *automation of reasoning* is one of the major *intellectual achievements* of computer science.

(5) Interactive *proof assistants* are creating a worldwide formal digital library of mathematics, computing theory, and verified programs.



3 Technical components of the course

Logical System Name	**Computational Core Name**
Propositional Calculus (PC)	Intuitionistic PC (IPC)
First-Order Logic (FOL)	Intuitionistic FOL (iFOL)
Peano Arithmetic (PA)	Heyting Arithmetic (HA)
Type Theory	Intuitionistic Type Theory (ITT)
	Constructive Type Theory (CTT)

Each of these mainstream widely studied logics has a *computational core* and a classical variant/extension. We will start the study of each case by looking at the computational core and relating it to computing theory, programming logics, formal methods, and programming practice.

Logical Systems from Projects

Some of you may also study logical systems in your projects such as Set Theory (computational cores are Intuitionistic Zermelo-Fraenkel (IZF) and Constructive ZF (CZF)). You might also study Hilbert's Geometry (HG) or Tarski's Geometry (TG). You might study Event Logic or programming logics such as Hoare Logic.

The computational core of all these logics is a *domain specific programming language*. We could name them also by the logic name IPC-PL, iFOL-PL, HA-PL, CTT-PL. The theories CTT and ITT contain implemented Turing complete program languages. Only CTT has a logic of these partial recursive functions.

4 What are the major results we will cover?

- Completeness of PC and IPC
- Gödel's completeness theorem for FOL
- CB11 Completeness theorem for iFOL
- Church, Turing undecidability results
- Gödel's incompleteness theorems
- Programs-from-Proofs and correct-by-construction programming
- Embedding into the computational core
  FOL into iFOL
  PA into HA
- Logic of Events and reasoning about distributed processes