First-order logic (FOL) has been thoroughly investigated in the last 130 or so years, and it is a very well understood logic. FOL gains its importance from its wide applicability and from its place in the landscape of logical calculi and formal languages.

Some elements of classical logic are studied in Phil 120, which is a prerequisite for this course. (The prerequisite can be waived upon request in certain cases.) This course is a more detailed and more formal study of some of the same topics that were touched upon in Phil 120, together with new and more complex questions and methods from FOL and beyond. For example, in this course, we will look at some first-order theories, which are concrete applications of FOL.

The course intends to develop and advance your understanding of FOL, its building blocks and their interactions. Some of these components are the truth-functional connectives, the quantifiers, the identity predicate, as well as proof systems and models. The course will enhance your ability to formalize some subtle and crafty natural language sentences. Toward the end of the term, you will have a chance to learn about resolution, which is a proof system for FOL that is widely used in computer science applications, and to take a glimpse at induction and set theory.

We will use a textbook that was written by world-class logicians who aimed at providing an excellent text and superior tools for learning logic. The textbook is accompanied by a software package that contains Boole, Fitch and Tarski’s world. (These three programs are named after famous logicians; a fourth program called Submit can facilitate your getting rapid feedback.)

- Boole makes easier and speedier the construction of truth tables.
- Fitch is an implementation of the so-called Fitch-style natural deduction system. Using this program you can prove theorems of first-order logic in a rigorous way.
- Tarski’s world allows you to construct a small model consisting of blocks, in which you can evaluate sentences. You can vary the model and the sentences, and you can play games with Tarski’s world to clarify and visualize the truth condition of a sentence in a concrete model.

These programs provide a lot of opportunity for experimentation, exploration and learning. (No programming or computer science experience is required for success in this course.)

Time: M, W, F 10:00 am–10:50 am


For further information, please contact the instructor at <bimbo@ualberta.ca>

The (official) course outline is available in the e-classroom during the course.