# CS 611

## *Objectives*

The core objectives for CS 611 are…

1. Confidently read and write proofs in or for the CS literature.
2. Understand foundational definitions and results in complexity theory
3. Understand foundational definitions and results in computability theory
4. Correctly interpret complexity and computability results as they relate and don't relate to research.

An important delimitation for CS 611 is that it is not a class that prepares students for research in theoretical computer science.

## *Competencies*

Some specific competencies for CS 611 are…

Reading and writing proofs:

- Construct and deconstruct a logically sound argu$\phi$ment based on the structure of the claim to be proven.
- Discern between a bad proof and a good, but difficult, proof.

Computability theory:

- Godel encodings
- Total and partial computability and ennumerability, Turing acceptability
- Self application, diagonalization,
- Classical and intuitionistic logic as it relates to decidability.
- smn theorem,
- Greatest fixpoints in the Recursion theorem
- Rice's theorem,
- Many to one reducability and decidability.
- Oracle TMs and computability hierarchy.

Complexity theory

- Little-oh and big-oh notation
- Space compression, linear speedup
- Definition and value of time/space constructable functions.
- complexity classes and their inclusion and separation results.
- Savitch's theorem (NPSPACE = DPSPACE)
- Space and time hierarchy theorems.
- Use of polynomials to divide "feasible" and "infeasible"
- Consequences of P=NP. Consequences of P!=NP
- Turing and many-to-one reductions to define C-complete and C-hard classes.
- Proving membership in NP-Complete. What that implies and doesn't.

Other things

- role of computability and complexity theory in computer science.

## *Assessment*

1. [Read and write proofs]
    a. Semester project involves writing a proof from their research area suitable for publication. During the semeseter, we simulate a review process in which students submit, review, revise and resubmit their proof projects. Students are graded on their submissions and their reviews of other students submissions.
    b. First part of the semester involves writing proofs for homework. These are graded for completeness.
    c. The reading in complexity and computability theory require students to reverse engineer most proofs in the textbook throughout the semester.
    d. Students will write proofs on the final exam.
2. [Understand foundational definitions and results in computability theory]
    a. Reverse engineering of proofs.
    b. In class discussion to informally assess understanding
    c. Final exam
3. [Understand foundational definitions and results in complexity theory]
    a. Reverse engineering of proofs
    b. In class discussion.
    c. Final exam
4. [Apply theoretical results to their research]
    a. Three short papers in which students find and evaluate proofs in papers in their research area, find complexity results and their impact in their research and assess the impact of a computability result in their research area.
    b. The semester proof project.