Course description:
This course covers the fundamentals of translation including an introduction to the structure and organization of programming languages. We review regular expressions and context-free grammars. Within the fundamentals of translation, translating techniques, e.g., LR, LL, and recursive descent are studied. We study syntax and lexical analysis, symbol tables, semantics and parsing, and code generation. Prerequisite: CSS 343 or CSS 502 (or equivalent content).

Learning objectives:
At the end of this course, the student will be able to
- Define compiler theory and, in general, translation of programming languages terminology
- Describe the techniques used in translation and compiling (LL, LR, recursive descent)
- Explain the underlying theory of these techniques and how they differ
- Explain and implement the lexical analysis phase of a compiler
- Write a context-free grammar and implement the parser phase of a compiler
- Design and implement a symbol table for a compiler
- Implement code generation built on the lexical analyzer, symbol table, and parser
- Demonstrate in-depth knowledge of an advanced compiler topic (548)

448 Grading:
- Project, homework: 55%
- Midterm exam: 25%
- Final exam: 20% A scale of 90, 80, 70, 60 will be approximately followed.

548 Grading:
- Project, homework: 55%
- Midterm exam: 20%
- Final exam: 20%
- Research paper/presentation: 5% A scale of 90, 80, 70, 60 will be approximately followed.

Research paper (548):
You are to write a 1500-word paper on a compiler topic of your choice. The intended audience is another student in this course, someone who will benefit from an in-depth discussion of one aspect of compiling. You will give a five-minute presentation on the topic.

Textbook:
A compiler book is recommended, but not required. Two good ones that I primarily use:
- Engineering a Compiler, K. Cooper and L. Torczon, Morgan Kaufmann
**Policies:**
You will work with a partner on the compiler project. You and your partner must document who does what on the project.

All work other than the project must be done independently. Any collaboration of work will result in severe penalty. You may discuss a homework problem statement and any clarification with each other, but any actual work to be turned in, must be done without collaboration.

All assignments are graded on correctness, readability, and good organization. Code is additionally graded on clear and complete documentation, efficiency (no unnecessarily complicated or unnecessary code), good style, clarity and format of output, and following instructions.

Any homework is due at the beginning of class on the due date. It may be turned in at the next lecture for a 20% reduction in grade but not thereafter unless prior arrangements have been made.

To request academic accommodations due to a disability, please contact Disabled Resource Services (DRS) 425.352.5307, 425.352.5303 (TDD), 425.352.3581 (FAX), or at drs@uwb.edu. The DSS counselors will contact me and we can discuss accommodations.

**Course goals:**
This course is designed to provide you with an understanding of the terminology, techniques, and underlying theory in the translation of programming languages. The overall goal of CSS 448/548 is to take you through the translation process, both theoretically and practically. Translation is from one language or representation to another. Many different techniques to accomplish this are studied and the most powerful, one that would be used for a compiler, is implemented. Good software engineering and algorithm analysis techniques are used throughout. As with most technical courses, it takes time to learn and master the subject. Expect to spend an average of 10 to 15 hours a week outside of class time on this course.