Course Code: CSTE 4109 **Course Name:** Compiler Construction

Book:

Compilers: Principles, Techniques, and Tools by Alfred V. Aho, Ravi Sethi,
Jeffrey D. Ullman

Welcome to the Compiler Construction course. In this course we are going to discussing about, how programming languages works and implementation of Programming Language.

There are two major approaches of implementing Programming Language.

- Compilers
- Interpreters

In this course we are going to discuss about compilers.

We will cover the following Topics:

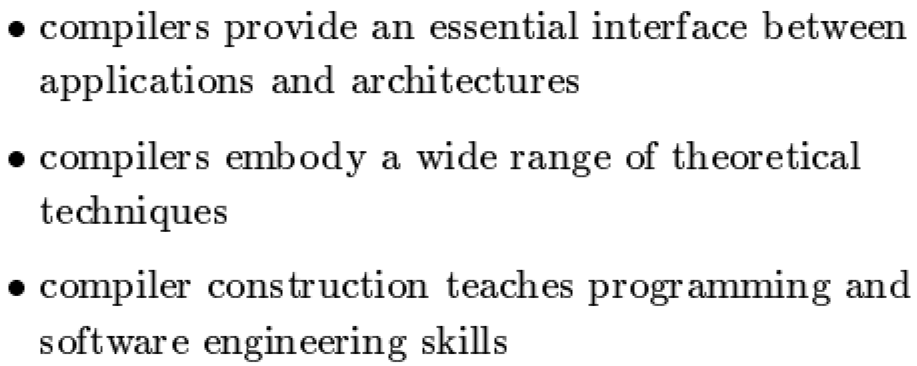
1. Introduction
2. Lexical Analysis (Scanning)
3. Syntax Analysis (Parsing)
4. Syntax Directed Analysis
5. Symbol Tables
6. Code Generation
7. Code Optimization

Prerequisites:

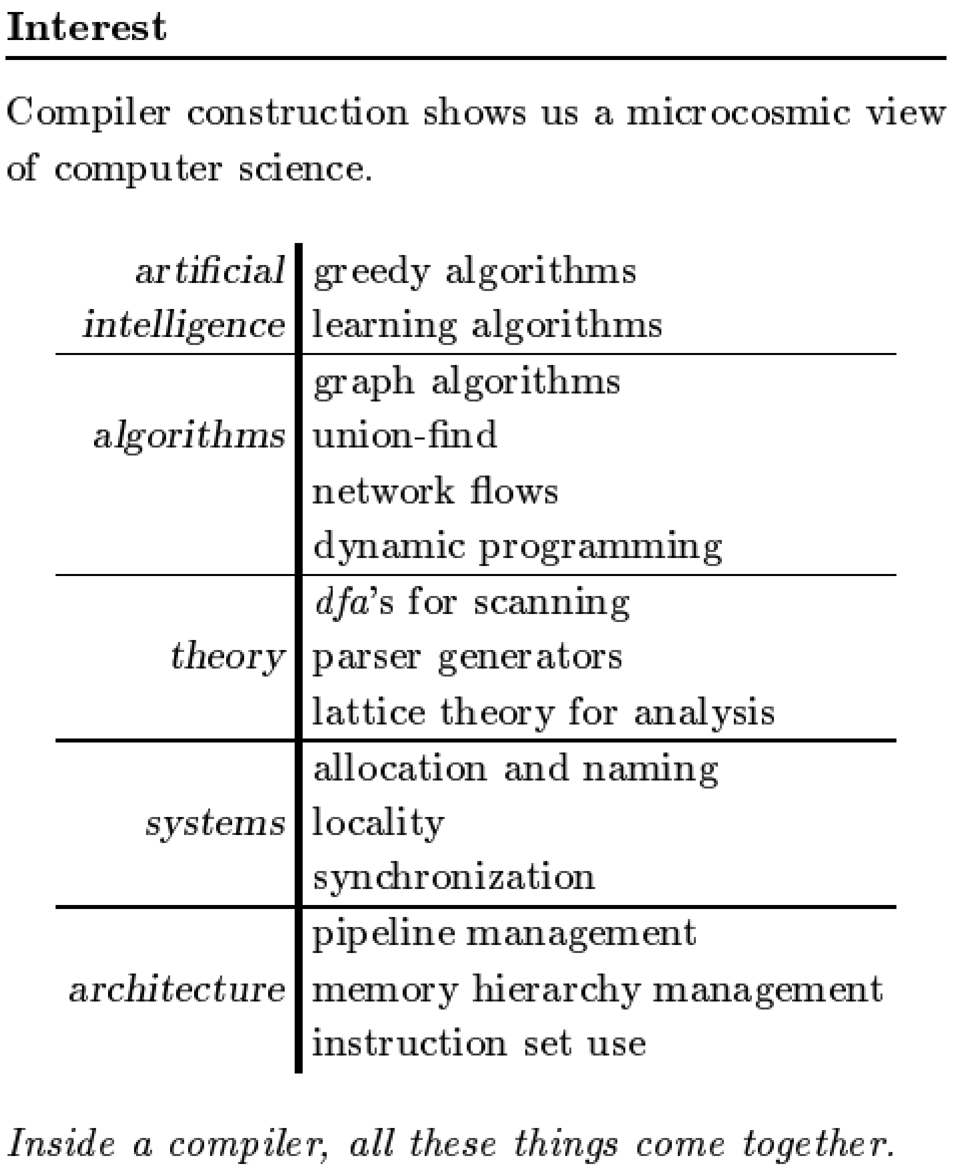
1. Good knowledge in C++. Knowledge in UNIX will be a plus.
2. Good knowledge in Theory of Automation
3. Knowledge about Browsers. (I recommend Netscape)

If you are not adept in those subjects or things, I suggest you to be prepared to spend a lot of time on studying those things. Be prepare to spend several night without sleep. If you want to do well in this course I suggest you to maintain a regularity and complete your assignment in time.

Why we should study and learn Compiler Construction?



- compilers provide an essential interface between applications and architectures
- compilers embody a wide range of theoretical techniques
- compiler construction teaches programming and software engineering skills



Interest

Compiler construction shows us a microcosmic view of computer science.

artificial intelligence	greedy algorithms learning algorithms
algorithms	graph algorithms union-find network flows dynamic programming
theory	*dfa*'s for scanning parser generators lattice theory for analysis
systems	allocation and naming locality synchronization
architecture	pipeline management memory hierarchy management instruction set use

Inside a compiler, all these things come together.

So let's be ready for making some fun and learn some interesting things!