

COP5621/CIS4930 Compiler Construction
Course Overview
Spring 2020

A translator is a program that has the capability to read a program written in a source language and translate it into a semantically equivalent program in a target language. For a compiler the source language is a high level programming language and the target language is assembly language or machine code. All of you have used a compiler. Now you will get an understanding of how this magical black box actually works.

We will be studying all of the major phases of the compilation process. This includes lexical analysis (scanning), syntactic analysis (parsing), semantic analysis (semantic checks), intermediate code generation, target code generation, and optimization. Knowledge of this material may be quite helpful for a number of other applications, which include interpreters, editors, static analyzers, debuggers, text formatters, and word processors.

The text we will be using is "Compilers: Principles, Techniques, and Tools" by Aho, Lam, Sethi, and Ullman (or also known as the dragon book). Most faculty who teach a compilers course consider this text to be the definitive text on compilers.

The only way to really understand a compiler is to implement one. You will be implementing a number of assignments, one for each of the different phases of a compiler. Each assignment involving a specific phase will either be small and self-contained or I will give you a portion of the code and you will complete it. The final result will be a series of programs that takes programs written in a large subset of the C programming language as input and produces a SPARC assembly language program as output, which can be assembled and executed on the department's *program* stack.