# Computer Science 160
# Translation of Programming Languages

Instructor: Christopher Kruegel

# CS 160 Info



- Web page: https://sites.cs.ucsb.edu/~chris/teaching/cs160/index.html
- Discussion board (Piazza)
  - class page: https://piazza.com/ucsb/fall2023/cs160/home
  - signup: https://piazza.com/ucsb/fall2023/cs160

# Requirements



- The course requirements include
  - several projects
  - a midterm and a final exam
- The projects (and exams) are individual efforts
- The final grade will be determined according to the following weight
  - projects: 50%
  - exams: 50%

# Compiler Project



- Goal of entire project is to build a compiler
- Sub-projects cover all parts of compilation process
  - Read in and parse code
  - Check code for semantic properties (type checking)
  - Generate executable x86 machine code
- Compiler will work on a simple, C-like language
- You will use well-known tools and some C++ code that we provide

# Material



- The course will adopt the following book:

  Keith D. Cooper and Linda Torczon

  Engineering a Compiler (EaC) – 3rd edition

  Morgan Kaufman (Elsevier)

- The set of assignments will be updated during the course

- Additional material is provided on the class Web page

- Questions to instructor, TAs and fellow students via Piazza

# What you will learn in this class?



- High-level understanding of the steps involved in compiling a program down to something a machine can read

- In-depth understanding of some of the interesting parts of the compiler: For example, how is your code parsed and how is machine code generated?

- Understanding of what needs to be done to read and write complex input and output formats (such as program source code, XML, Java bytecode)

# Topics



- Overview of compilers
- Lexical analysis (Scanning)
- Syntactic analysis (Parsing)
- Intermediate representations
- Type checking
- Symbol tables
- Code generation
- Runtime environments
- Code optimization (if time permits)

# Why take this class?



- Compilers are a testament to the power of computer science
  - Theory, algorithms, systems, architecture… all these things you practice in other classes are **applied** to compilers!
- Bridge a huge mental gap between the software you know how to write and the hardware you know how to build
- The techniques you learn in this class are applicable to many real-world problems you may face "on the outside"
  - Input and output parsing (XML)
  - Application specific languages (configuration files)
  - Program analysis and understanding

# Expectations



**Students**

Check the webpage often

Check Piazza often, and before emailing a question

Answer questions in class

Stay on top of your work

**Please go first to TA for all project questions**

If you want a re-grade, re-grade will be for the full assignment/test

Know the vocabulary of compilers

**No Cheating**

**Instructor and TA**

Announcements will be posted on Piazza

Check email/posts and respond promptly

Be available before tests or project deadlines

Return exam and project results promptly

Take cheating seriously