

CS 375: Compilers

A computer is:

- a fast Pocket Calculator
- with a large Memory

One second of computer time would take you the rest of your life to duplicate with a pocket calculator.

Converting from a language such as Java to the pocket-calculator machine language (the *only* executable form) is the task of the compiler.

Programming Projects

We write lots of programs in CS 375:

- Lexical Analyzer: Characters \rightarrow Words (Tokens)
- Lexical Analyzer using `lex`
- Parser: Words \rightarrow Sentences (Trees)
- Code Generator: Trees \rightarrow Machine Code

The programs are in Lex/Yacc/C (most common), C, or Lisp.

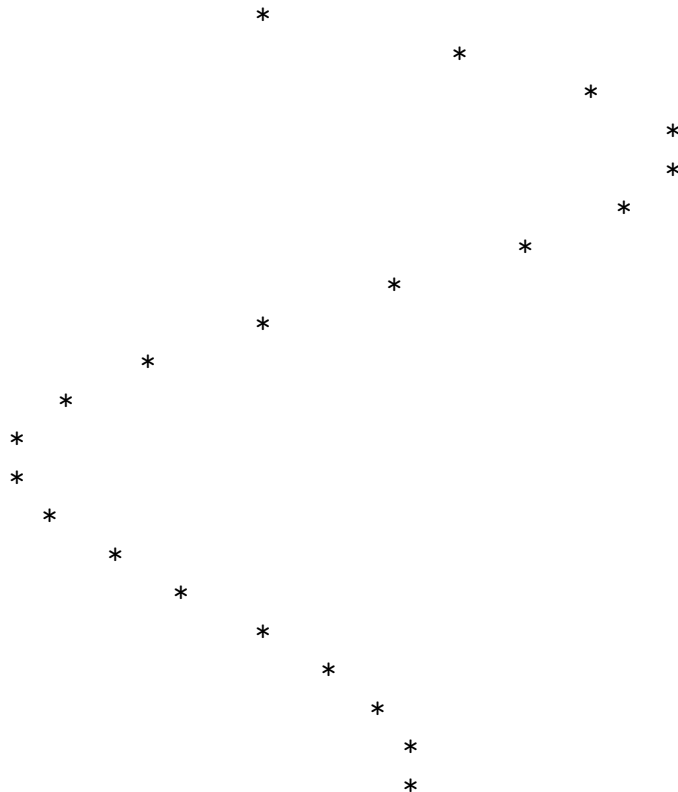
These projects combine to form a real compiler from Pascal to X86 machine code that executes on hardware.

Pascal Test Program

```
{ program 4.9 from Jensen & Wirth: graph1.pas }
```

```
program graph1(output);
const d = 0.0625; {1/16, 16 lines for [x,x+1]}
      s = 32; {32 character widths for [y,y+1]}
      h = 34; {character position of x-axis}
      c = 6.28318; {2*pi} lim = 32;
var x,y : real; i,n : integer;
begin
  for i := 0 to lim do
    begin x := d*i; y := exp(-x)*sin(c*x);
          n := round(s*y) + h;
          repeat write(' '); n := n-1
            until n=0;
          writeln('*')
    end
end.
```

calling graph1



Why Take CS 375 ?

Capstone: CS 375 Combines All Areas of CS

- Data Structures
- Programming Languages
- Theory
- Architecture

CS 375 Makes You a Better Programmer

Understanding how high-level language constructs map to hardware provides a deep understanding of efficiency. You will also understand type systems and language features better.

Grad Schools and Employers Want CS 375

Students Say CS 375 Changed Their Lives