EECS 368
Programming Language Paradigms

Dr. Andy Gill

University of Kansas

August 25, 2015

Class Information

How to find me

Office:	2024 Eaton Hall / 252 Nichols Hall
Web:	`http://people.eecs.ku.edu/~andygill`
Phone:	4-8817 / 4-4712
Email:	`andygill@ku.edu`
Office Hours:	2:00-4:00 Wednesdays in Eaton Hall, or by appointment.

About the Class

Time:	4:00-5:15 TR
Class Web:	`https://piazza.com/ku/fall2015/eecs368/home`
Prerequisites:	EECS 268 is a hard prerequisite for this course.

No Class Texts

- Friedman and Felleisen, The Little Schemer, MIT Press, 1995, 4th edition. ISBN: 0262560992
- Hutton, Programming in Haskell, Cambridge University Press, 2007. ISBN: 0521692695
- Many other resources on class website.

Who is Andy Gill?

- Researcher in the field of programming languages
- Ph.D. from Glasgow University
- Spent 12 years in industry as a compiler engineer
    - Worked on a Java compiler, a C++ compiler, and a Haskell compiler
    - Almost accepted a job working on Firefox's JavaScript compiler
    - Co-founded a technology startup in Portland, OR
- Want to share my research interest: programming languages

Back in the 50's

Machine Codes

Programing computer by literally giving the codes to perform operations. Examples are

- moving data (0x37)
- adding data (0x17)
- comparing data (0x28)
- storing data to tape (. . .)

This was interacting with the machine on its terms, 1s and 0s.

Using Software Abstractions

Assembly Language

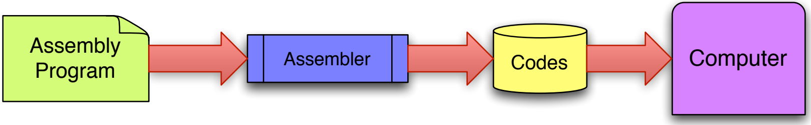
Programing computer by using mnemonics instead of numerical code. Examples are

- moving data (ld r1,r2)
- adding data (add r1,r2,r3)
- storing data to tape (cmp r1,r2)
- goto to another set of instructions (goto label_44)

This is slightly better.

- A transliteration that is easier for humans to understand/remember.
- Still a one-to-one mapping to machine code.

What an Assembler Does


Assembly Program
Assembler
Codes
Computer