EECS 368
Programming Language Paradigms

Dr. Andy Gill

University of Kansas

January 20, 2017

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 Wed in Eaton, or by appointment.

About the Class

Time:	10:00-10:50 MWF
Class Web:	`https://piazza.com/ku/spring2017/eecs368/home`
Prerequisites:	EECS 268 is a hard prerequisite for this course.

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

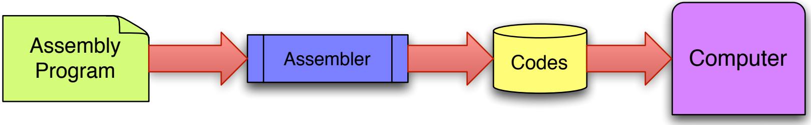
Programing computer by using mnemonics instead of numerical code. Examples are

- 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



Using Hardware Abstractions

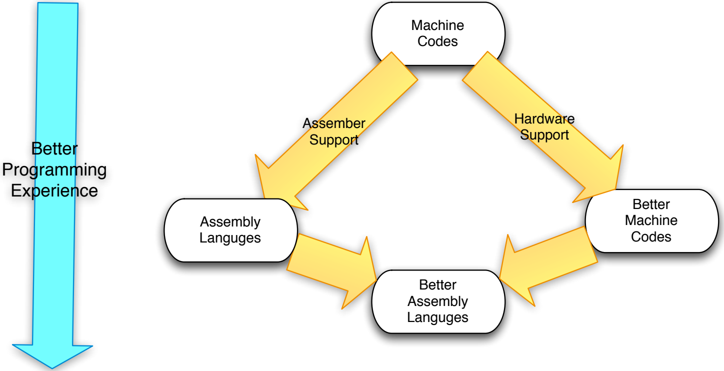
Better Codes

At the same time, the codes provided inside machines became more powerful

- codes for subroutines
- codes for multiplication, division, floating point.
- . . .

Can do the same operation faster and better.

Narrative of Early Computer Languages



A Short and Edited History of Computer Languages

((Whiteboard))

Other Languages

There are many other idioms and ideas in programming languages.

- Different computational models: Logic languages, SQL, XSLT
- Scripting languages: TCL, Perl, Python
- Web services language: PHP, JavaScript

In this class, we are going to focus on the two main computational models, imperative and functional.