# Lecture 02 - Layers of Representation in Systems

## Thinking in categories

“If the only tool you have is a hammer, then everything looks like a nail.”

Colors and categories.

4 14 23 34

Programming Example: Merkel Hashes - Lots of uses. BlockChain is one. Mike did a class where the implementation was about 1000 lines of code. I went and looked at implementations for my BlockChian class - 1140 to 1521 lines of code. But wait…. To build a blockchain you only need a couple of features - and if you take advantage of the machine architecture - in my class 32 liens of code. Performance better under every test case!

This is the first class that really will start to pay long term dividends.

Anything you can do in software you can do in hardware and…

Anything you can do in hardware you can do in software.

The only real differences are speed and cost.

This class is about looking at computers from many different points of view and about the “cost” and “speed” of different options.

## Computer Representation in Hierarchies

1. Highest level - Garbage in Garbage out.

“Any sufficiently advanced technology is indistinguishable from magic.” Arthur C Clarke’s First Law

1. Self driving / Google search / Bots / Medical Diagnosis / Spying on you with your Phone / Doing a really bad job of voice recognition - Computers can do anything? No… Actually what they can do is really really limited and stupid.

| Instruction | Description |
|---|---|
| Load | Load a value(number) from a memory to a register |
| Store | Store a value back to memory from the register |
| Increment | add 1 to the value in the register. |

| Instruction | Description |
|---|---|
| Decrement | subtract 1 from the value in the register. |
| Branch if Not zero | Loop to some other location in the program when register is not zero. |

Sleeping at the wheel:

Tesla’s are safer than humans at driving. 2.1 billion miles of Auto-Pilot! Waymo (Google/Alphabet)’s self driving 39 million miles without a drive at all! 3 non-injury accidents that were not their fault!

## The Layers of Representation

Layers: - Applications - High level languages - Operating System - Machine Language - Microcode | Direct Hardware - Logic circuits - Gates - Transistors | Resisters | Capacitors - Silicon structures (Quantum Holes)

### What most persons see

1. User Interface (input)
    - web
    - gui
    - command line
    - binary input to a computer
    - electricity / on / off
2. User Interface (output)
    - reports
    - printouts
    - control of systems
    - screen

## A Single Programming Language

1. The “categories” - the “model” that the program uses. Von Neumann Fetch Execute
2. Imperative is the most common
3. Object oriented
4. Functional
5. Concurrent (real concurrency) - Microcode instructions
6. DFA - Deterministic Finite Automata
7. Push-Down Acceptor: LALR(1)

Inputs change “state” in a program - that then results in the possibility of new “state”.

## Multiple Languages Interacting

HTML on a web page, Cascading Style Sheets (CSS), with JavaScript - talking to a remote system in some language like Python and SQL (database).

## Simple C Program

```
#include <stdio.h>
int main() {
        putc('A',stdout);
}
```

1. System software like a “compiler” that turns our “language” into what the machine can run.
2. An Operating system to run the program.
3. Connection for the program to input/output.

4. Memory.
5. A loader and a file system.
6. Load - and Run The Code.
7. Display the output.

## Assembly Language Code

```
/ Program to output 'A' as a number
        Load X
        Output
        Halt
X,      DEC 65
```

And in Hex we get 0041 - is that an ‘A’? Why?

ASCII

Representation

What is “HEX”.

There are 10 kinds of people. Those that understand binary and those that don’t. Either you already understand binary or you are going to before the end of this class.

## Binary Representation of Program

```
1003
6000
7000
0041
```

## Electrical Representation of Program

Memory

Layers of Storage

Gates

Microcode Architecture

Boolean logic