PROGRAMMING MERIT BADGE 2018

Let's see just how far the rabbit hole goes!
<code src="slides"></close>

Instructor Info

- Chris Jones
  - Merit Badge Counselor for Programming
  - Committee Chair, Pack 439
  - Bear Den Leader, Pack 439
  - Contact Info
    - Cell: 509-979-9721
    - ccjones007@gmail.com

Chris' Career

- Master in Computer Science from WSU
- Started career as Research Scientist @ BBN Technologies
- Became a backend/server developer
  - Worked at TriGeo Network Security for 7 years
  - Solarwinds for 3 years
- Research-QA Engineer at Tenable Network Security
- Software Engineer at Tenable Network Security

Chris' First Computer

To talk about Software let's first talk about Hardware

What are the parts of a programmable device?

- Central processing unit
- Peripherals
  - Screen
  - Mouse
  - Keyboard
  - Hard drive
  - Memory
  - Touch screen
  - Compass
  - Printer

CPU
How does a processor work?

- Numbers, lots and lots of numbers!
  - Binary numbers
  - Two States – a 1 or a 0
    - What is a 1?
    - What is a 0?
      - Whatever we want them to be!
        - TTL logic – 5v or Ground
        - Modem – 1200Hz or 2000Hz
        - RS-485 protocol - which wire has a higher voltage

How does a processor work?
Continued…

- Two types of numbers
  - Instructions
  - Data
- Computer reads in an instruction and does what it is programmed to do when is sees that instruction
  - $A9 – Load the Accumulator
  - $80 – Store the Accumulator to memory
- Otherwise its just a number

Embedded Processors and Electronics Resources

- Suppliers
  - Adafruit.com
  - SparkFun.com
  - EvilMadScientist.com
- Learning
  - http://arduino.cc/en/Tutorial/HomePage
  - http://tronixstuff.com/tutorials/
- Project Ideas
  - http://www.instructables.com
    - Arduino and Raspberry Pi channels

In the end

That's just the hardware, if we want it to do something useful we need...

https://www.ted.com/talks/steven_johnson_how_play_leads_to_great_inventions

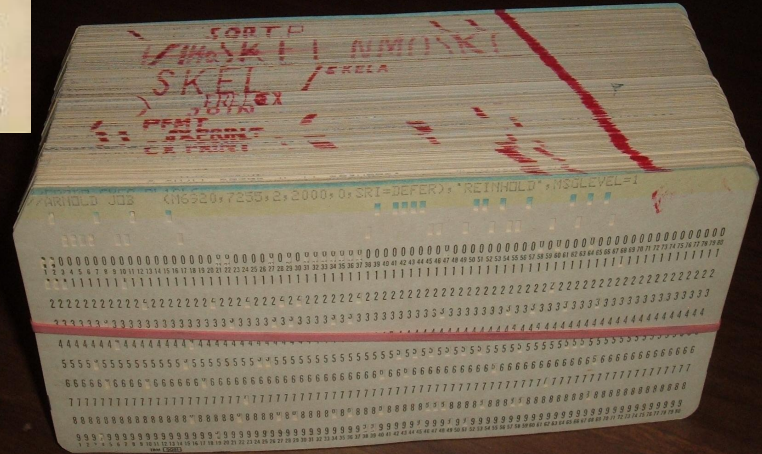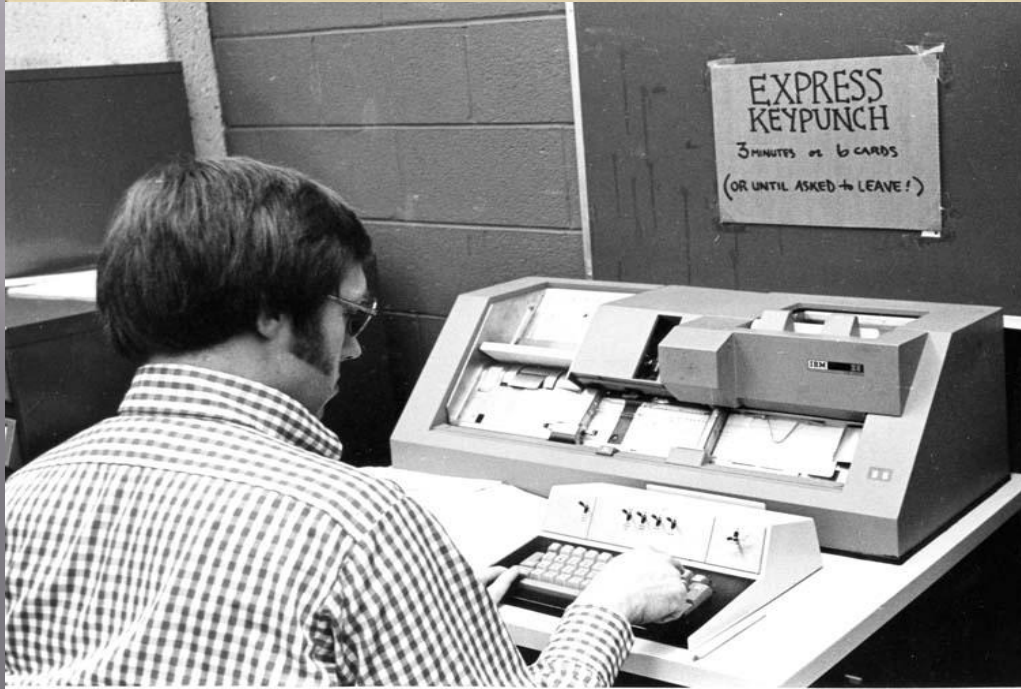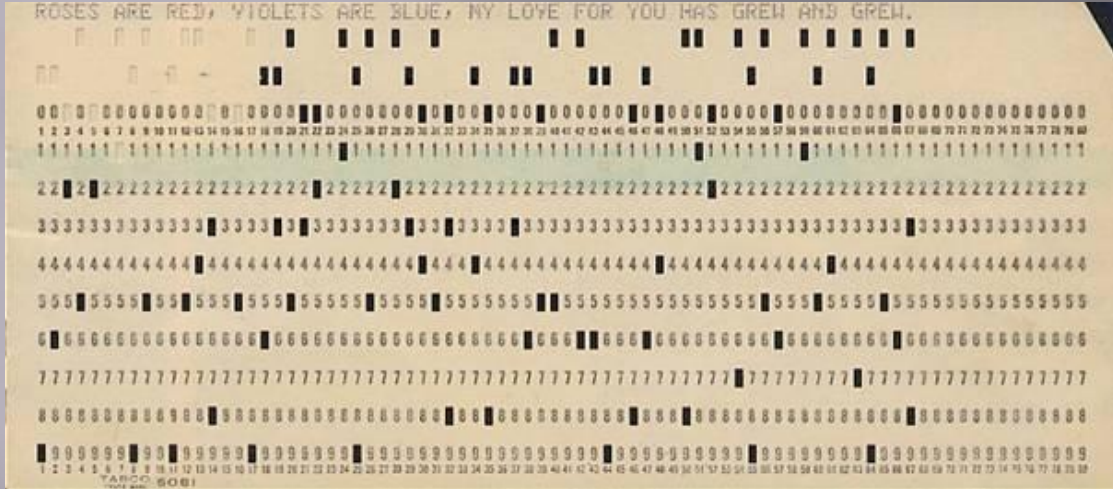# Devices that have code running

What is programming?

□ The process of developing and implementing various sets of instructions to enable a computer to do a certain task.

■ It is the glue that gets everything to work together

Code, Part 0001

Code, part 0010

History of Programming

One of the first home computers

Programming in the Dark Ages

- Switches and blinking lights
  - Entered the instructions and data into memory one byte at a time
- Paper tape
- Punch Cards
- Instructions go from the cards into the computers memory
  - Still what happens today, just faster and more convenient

First big Milestone

- Assemblers
  - Allowed programmers to work with a more human readable format
  - Managed memory to some extent
- Linkers
  - Allowed programmers to build reusable bits of code
  - Programmers could share code

Human Readable 2 Binary

```
section .text       ;section declaration
                    ;we must export the entry point to the ELF linker or
global _start       ;loader. They conventionally recognize _start as their
                    ;entry point. Use ld -e foo to override the default.
_start:             ;write our string to stdout

    mov edx,len     ;third argument: message length
    mov ecx,msg     ;second argument: pointer to message to write
    mov ebx,1       ;first argument: file handle (stdout)
    mov eax,4       ;system call number (sys_write)
    int 0x80        ;call kernel
                    ;and exit
    mov ebx,0       ;first syscall argument: exit code
    mov eax,1       ;system call number (sys_exit)
    int 0x80        ;call kernel


section .data       ;section declaration

msg db "Hello, world!",0xa      ;our dear string
len equ $ - msg                 ;length of our dear string
```

```
1111000101010111010101010101010111100001101010101010101101010101010101010100000001111111010101011111001110001010
1011100011010101010101010101011110001010101110101010101010101011100011010101010101011010101010101010101000000011
1111101010101111001110001010101110001101010101010101010101110001010101110101010101010101011100011010101010101011010
1010101010101010000000111111101010101111100111000101010111000110101010101010101010111000101010111010101010101010
1011110001101010101010101101010101010101010100000001111111010101011111001110001010101110001101010101010101010101
1110001010101110101010101010101011100011010101010101101010101010101010100000001111111010101011111001110001010101
110001101010101010101010111100010101011101010101010101011110001101010101010101101010101010101010100000001111
1110101010111110011100010101011000110101010101010101011100010101011101010101010101011110001101010101010101101
010101010101010000000111111101010101111100111000101010111000110101010101010101010
```

Second milestone
Compilers and Interpreters

- Led to the development of languages like C, Fortran and Pascal
- Very human readable
    - Printf("Hello World!");
- Allows a more expressive way of working

Object oriented programming

- Paradigm shift
    - Not how to do some thing
    - Describes a machine of parts and how those parts act
    - Each object has responsibilities and behaviors
    - Easier to maintain
    - Easier to modify

- Examples

Software!

And that takes programming!

Programming

- Is the glue that gets everything to work together
- Provides the flexibility to make the device do multiple tasks
  - Smartphones
    - Who makes calls anymore?
  - Hubble Telescope
  - Mars Rover

Scratch

- Visual Programming environment
- Developed at MIT to teach programming
- Great for developing games and animations
- Perfect for beginners
- Can interface with electronics through special boards
  - Makey Makey
  - Raspberry Pi
- http://scratch.mit.edu/

Alice

- Visual programming language
- Developed at CMU to teach programming
- 3D game creation
- http://www.alice.org/index.php

Javascript

- Designed to run within a web browser
- "Loosely typed" language
- With a number of new libraries, it is a great language for building thin clients within the browser

Python

- Interpreted Language – no compile step!
- Batteries Included
    - If you want to do something, there is probably a library to do it
- Dynamic language
    - Object properties and method can be created at runtime
- Available on almost any computing platform you can think of
- Used for all sorts of business applications and testing frameworks

C

- Compiled language
- Basis for a number of different languages
  - C++
  - C#
  - Java
- Can get as low level as assembly
- Used in embedded programming and systems programming
  - business and manufactoring applications

Arduino

- Microcontroller, not a computer
- Programmed in C from a computer
- Designed for interfacing with electronics
- Comes in lots of different variations
  - Uno
  - Micro
  - Explora
- Lots of libraries and examples online!
- Available at Radio Shack

Raspberry Pi

- A full on Linux computer
- Hooks up to a television
- Has some pins for interfacing with electronics
  - Not as many as the Arduino
- Can run any of the programming languages we have discussed

Questions?

Lets talk about code!

- [ ] Data
- [ ] Conditions
- [ ] Loops
- [ ] Code reuse / organization

Steps to writing Application

- Analysis
- Design
- Code
- Test

Where is Software used?

Programming Resources, 0001

- ☐ Our programming resources
    - ■ https://github.com/ccjones007/meritbadge
- ☐ Boys Life
    - ■ http://boyslife.org/programming/
- ☐ Codecademy
    - ■ http://www.codecademy.com
- ☐ Invent With Python
    - ■ http://inventwithpython.com/

Programming Resources, 0010

- Scratch
  - http://scratch.mit.edu/
- Javascript && HTML (write in webpage)
  - http://jsfiddle.net/
  - https://codepen.io/
  - http://js.do/
  -
- Various languages (write in webpage)
  - https://repl.it/
  - https://trinket.io/

Tools of the Trade

- Source version control
  - Software system to manage code base and updates
  - CVS, SVN, Git
  - https://github.com / https://bitbucket.org / https://gitlab.com
- Editor / Integrated Development Environmnet (IDE)
  - Eclipse for Java, etc., Visual Studio for C/C++/C#/etc.
- Tracking systems
  - Jira, Redmine, Bugzilla,
- Collboration tools
  - Wikis (MediaWiki, Confluence, Forums, etc.)

Questions?