

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
 - Class resources
 - <https://github.com/ccjones007/meritbadge>

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 (CPU)
- 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?

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 it sees that instruction
 - \$A9 – Load the Accumulator
 - \$80 – Store the Accumulator to memory
- Otherwise it's 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](https://www.ted.com/talks/steven_johnson_how_play_leads_to_great_inventions)

**Devices that have code
running on them?**

One of the first home computers



What is programming?

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

History of Programming



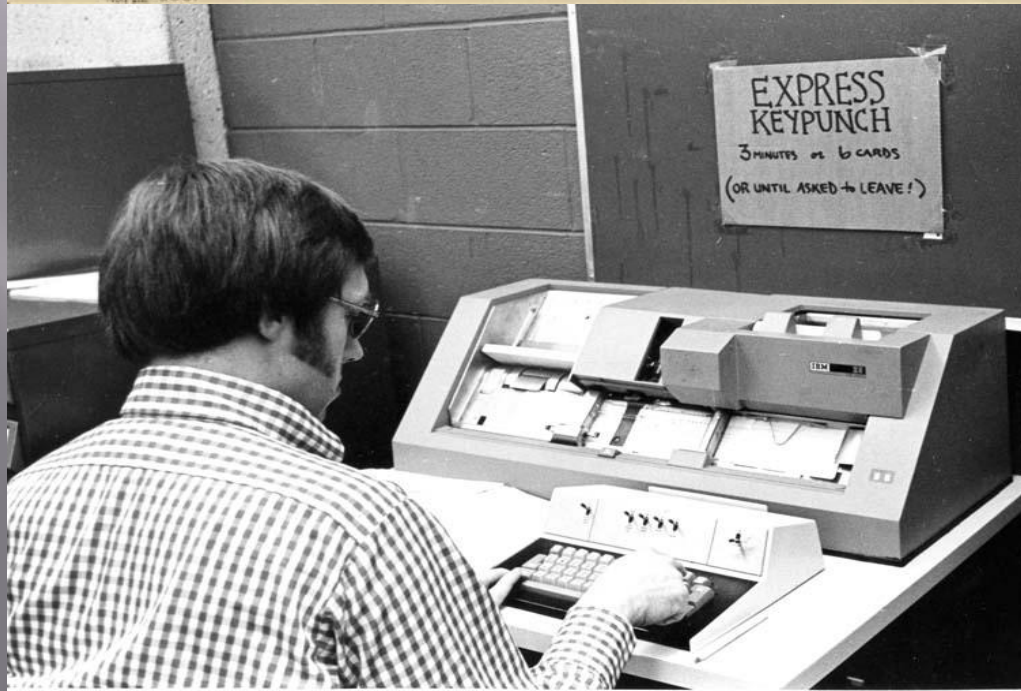
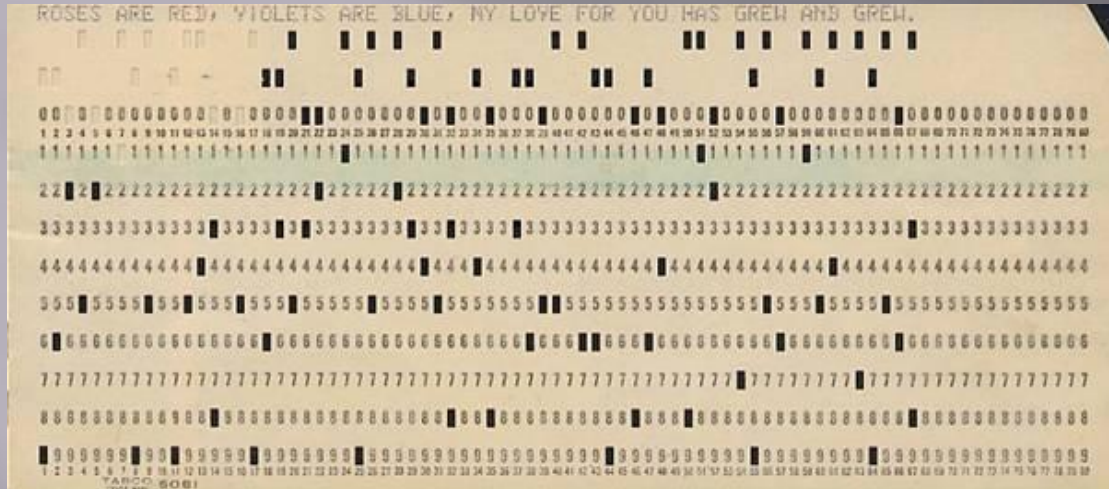
Code, Part 0001



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

Code, part 0010



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 .data      ;section declaration
```

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

1111000101010111010101010101010111100001101010101010101010101010000000111111010101011110011100010101
011100011010101010101010101011110001010101110101010101010101110000110101010101010101010101000000011
1111101010101111100111000101010111000110101010101010101010101110001010101101010101010101010110
101010101010101010000000111111010101011110011100010101011000110101010101010101110001010101101010101010
1011110000110101010101010101010101010101010000000111111010101011110011100010101011000110101010101010101
111000101010111010101010101010111000011010101010101010101010101010101010100000001111110101010111100111000101010
1110001101010101010101010101111000101010111010101010101011100001101010101010101010101010101010100000001111
1110101010111110011100010101011100011010101010101010101011100010101011101010101010101110000110101010101010101
010101010101010100000001111110101010111100111000101010110001101010101010101010

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

Programming Languages

Types of programming languages

- Procedural
- Functional
- Object Oriented

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

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 manufacturing applications

Java

- Object oriented
- Compile once, run anywhere
 - compiles to a intermediary set of instruction
 - runs in the Java Virtual Machine (JVM)
 - JVM are specific to Operation System/CPU
- web applications (eg. gmail)
- desktop application (programming tools)
- Android support for JVM language called Kotlin

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

Devices

Arduino

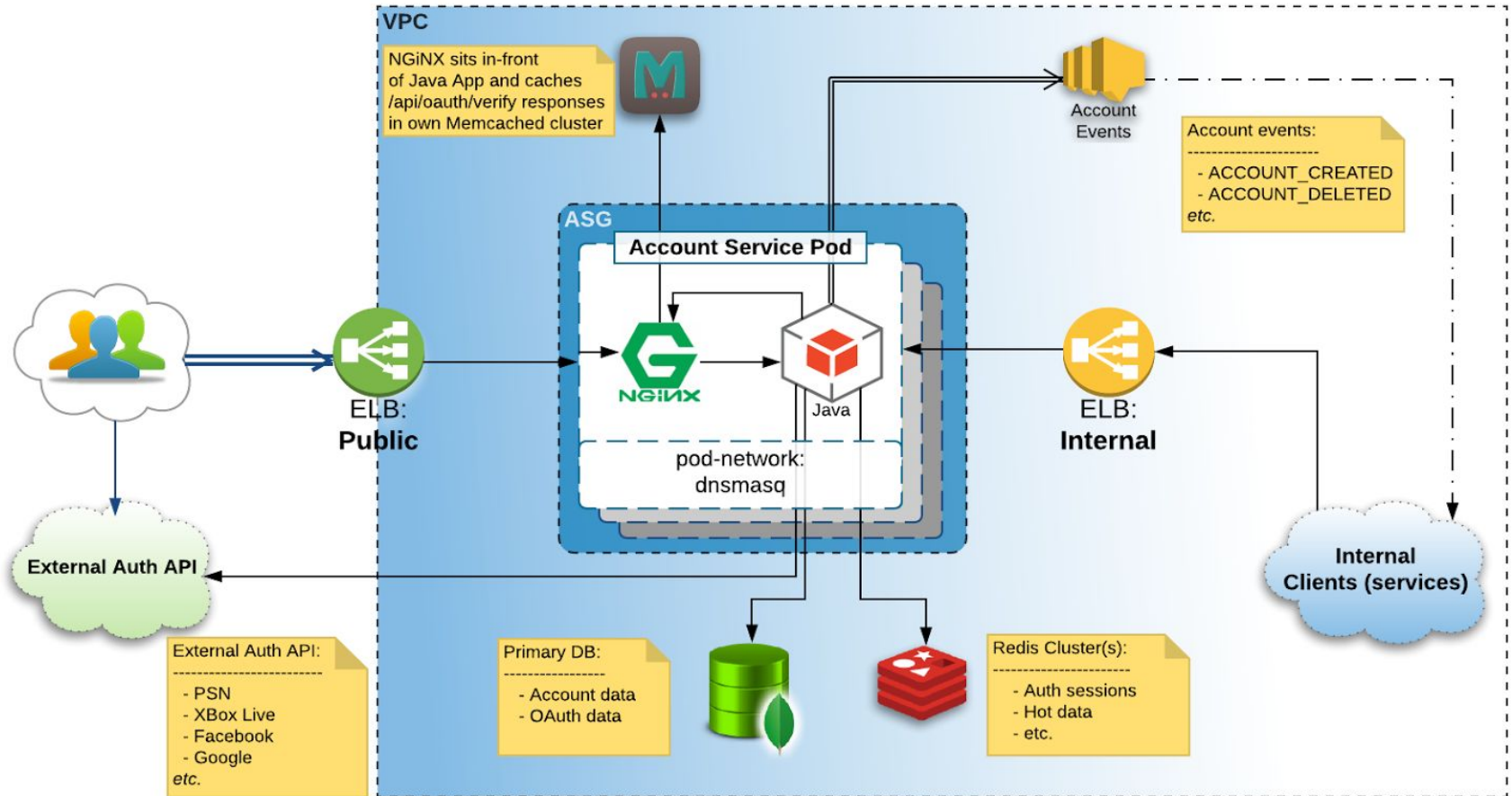
- Microcontroller, not a computer
- Programmed in C from a computer
- Designed for interfacing with electronics
- Comes in lots of different variations
 - Uno
 - Micro
 - Explora
- Lots of libraries and examples online!
- Available at Radio Shack

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?

Software System



<https://youtu.be/eQ4mvd7D87s?t=662>

Lets talk about code!

- Data
- Conditions
- Loops
- Code reuse / organization



Steps to writing Application

- Analysis
- Design
- Code
- Test

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 Environment (IDE)
 - Eclipse for Java, etc., Visual Studio for C/C++/C#/etc.
- Tracking systems
 - Jira, Redmine, Bugzilla,
- Collaboration tools
 - Wikis (MediaWiki, Confluence, Forums, etc.)

Questions?