Programming and You

Presented by Andrew Wintenberg, Tanner Hobson, and Kyle Seabash Parsley

P.S. The Java language is the best language.

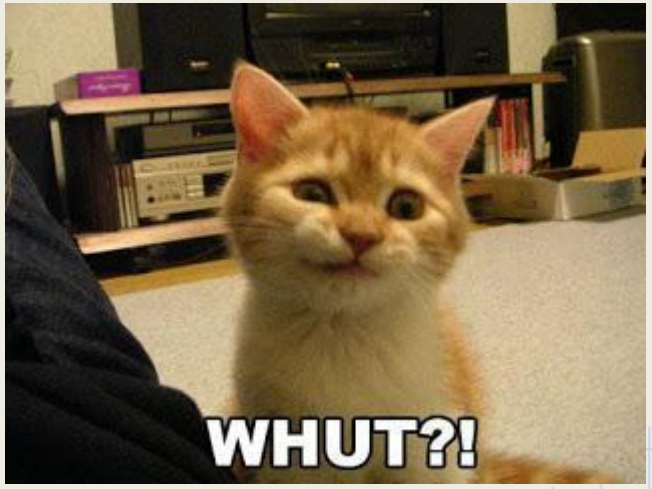
What good is programming anyways?

- With the expanding world of technology, many careers are emerging
- You can greatly simplify daily, repetitive tasks
- Robots
- Parentheses, Brackets, and Braces: Oh my!
- You can be an uber 1337 haxorz
- You can impress your friends with your mad tech skillz
- You can be the local tech wizard casting magic spells
- You can create a Visual Basic GUI to track the killer's IP address

First: What is Programming?

Programming is about:

- Variables
- Input and Output
- Encapsulation
- Turing complete template preprocessors (courtesy of C++)
- Tipstrips and Hotspots (courtesy of LabVIEW)
- AbstractSingletonProxyFactoryBean (courtesy of Java)
- Zygohistomorphic prepromorphisms (courtesy of Haskell)



Let’s try that again...

Programming is about a lot of things, and some of it is very complicated. Luckily, we only need to worry about a small portion of it.

Simply, programming is the art of teaching a computer how to perform tasks. Some like to refer to it as a series of transformations to some input, in order to get a desired output.

A Simple Example

- Imagine with me for a moment, that you are trying to instruct a computer to do a simple task, and you're only allowed 4 letters or less per word.
- We'll use the task of eating a lunch consisting of a bag of chips and a water bottle.

- This is a group exercise. Try to come up with some very specific instructions, with very specific cases
- Some words to get you started:

  - chip bag
  - open
  - put
  - make sure
  - H2O
  - see if
  - chew
  - sip
  - move down

Avoid general terms that encompass many steps such as "eat"

A Simple Example cont.

Some things you might not have realized:

- We have to check for errors a lot.
- “get chip out of bag” will fail if there aren’t any chips in the bag, so we have to check first.
- What if we start choking? Is there a way to *recover* from that error?
- What if the building you’re in catches fire?
- What if your friend calls your name?
- What if there’s a giant spider on your arm?
- Are there many steps to “open a bag”? “coordinate muscles in arm to move up and over until your hand rests on the bag. Coordinate fingers to grab one side of the bag. Coordinate other hand to meet the bag. Coordinate other fingers to grab other side of bag. Coordinate arms to move apart while ensuring your fingers stay gripped”, etc.

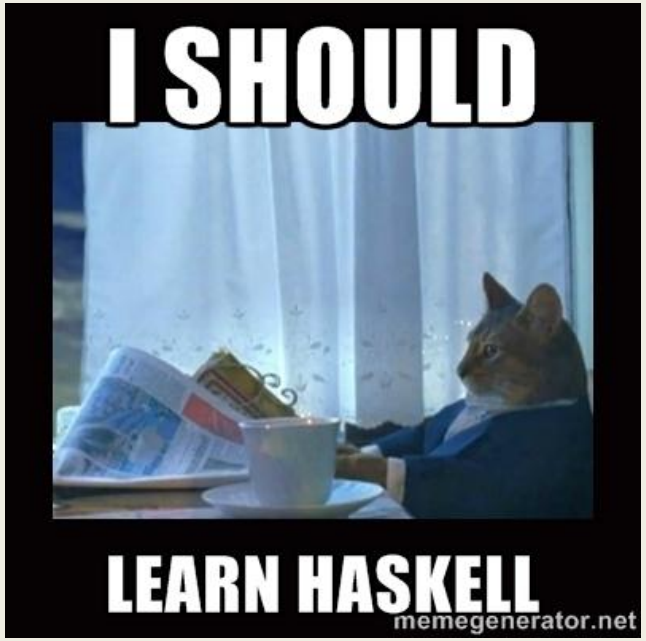
Types of Programming

- **List transformations** (APL/J/K, R)
  - If you have an operation, you perform it piece by piece, so that a list of the numbers 1, 2, and 3 added to 4, 5, and 6, would be 5, 7, 9
  - Useful for statistics

- **Object oriented** (Java, C++)
  - If you have an operation or feature you'd like to expose, you put it in a "black box" and only expose one interface, keeping everything neatly contained
  - Useful for enterprise software where you don't need to know how everything really works

Types of Programming cont.

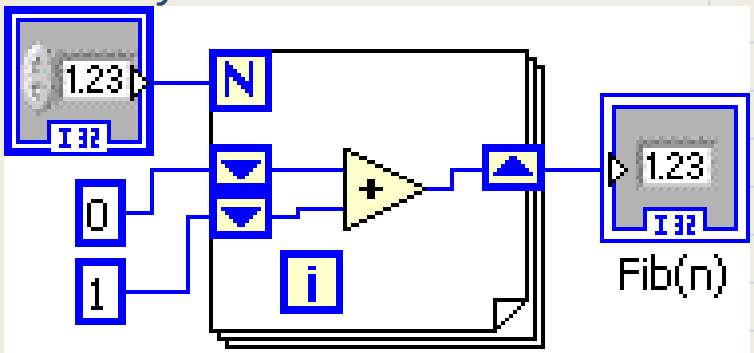
- **Functional** (Haskell, Mathematica)
  - If you have a large application, you first break it up into smaller pieces, and build up from there, similar to function composition in mathematics
  - Useful for mathematics, and applications where things change often



Types of Programming cont.

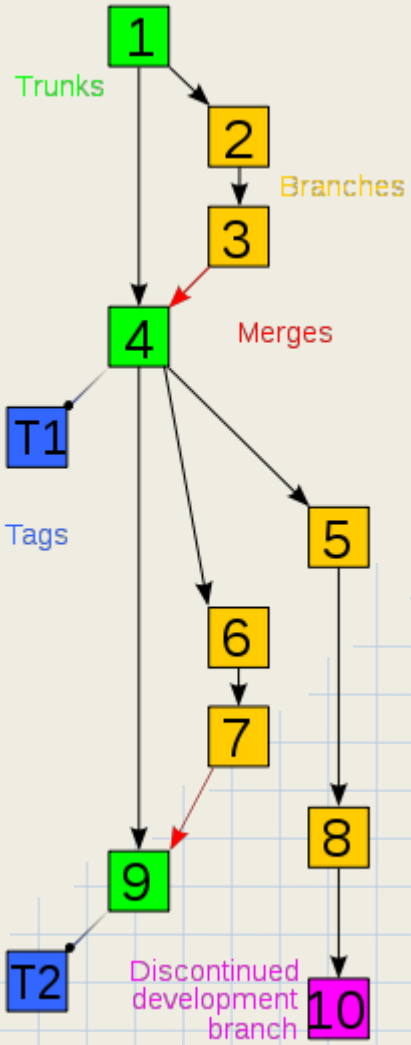
- **Logical** (Prolog, Sed)
  - You describe *what* the outcome should be, but not *how*, by defining a few rules and letting the computer figure it out
  - Useful for logic puzzles, artificial intelligence, and analyzing language

- **Graphical/Directed Acyclic Graphs** (LabVIEW, Flow.js)
  - These emphasize the flow of data through your program, and builds functions out of little “black boxes”
  - Useful for tracking data flow and building bigger programs out of small functions
  - Allows you to quickly prototype, and allows you to visualize the data coming from the robot quickly.
  - Graphical programming is usually used to show output quickly, such as output from the mars rovers.



Version Control Systems

- Version Control Systems (VCS) are programs that are used to keep track of changes and history in files
- Different applications control software in different ways, but it boils down to:
  - A folder is synced online (like Dropbox)
  - Changes are made locally on your computer
  - You describe those changes and push them online
  - If you ever mess up, you can just revert to a previous version
  - Multiple people can work at the same time and merge their changes

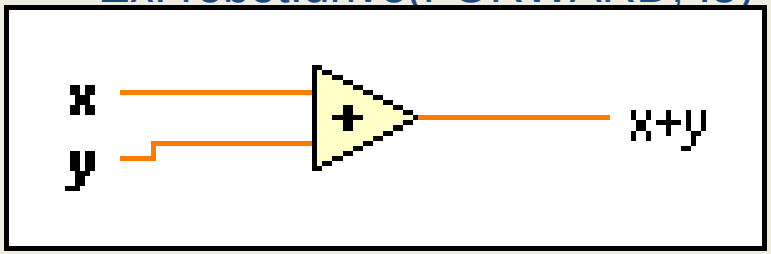


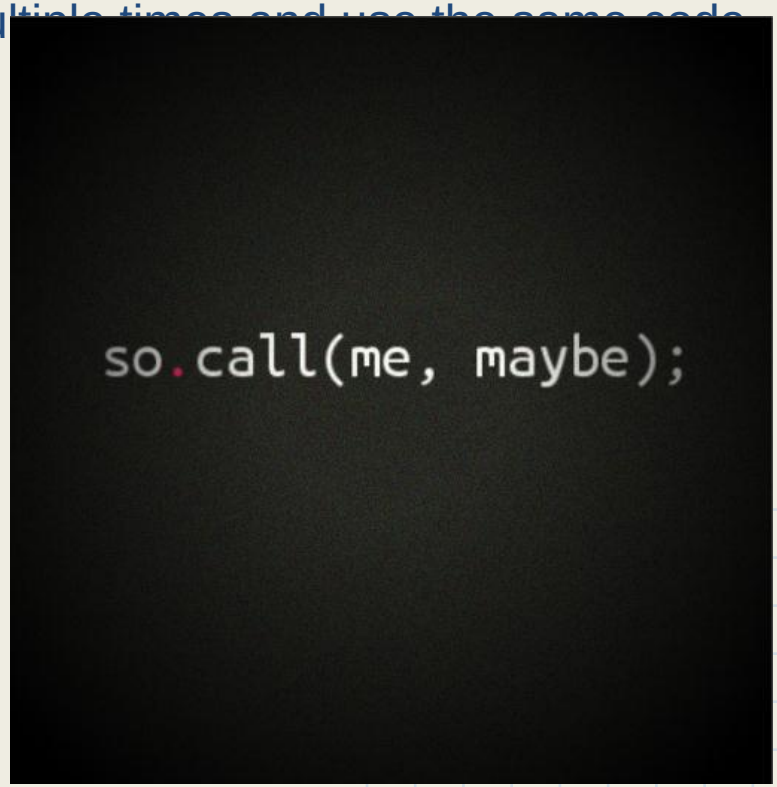
Programming Concepts

- Universal Concepts
- Language-specific concepts

Universal Concepts

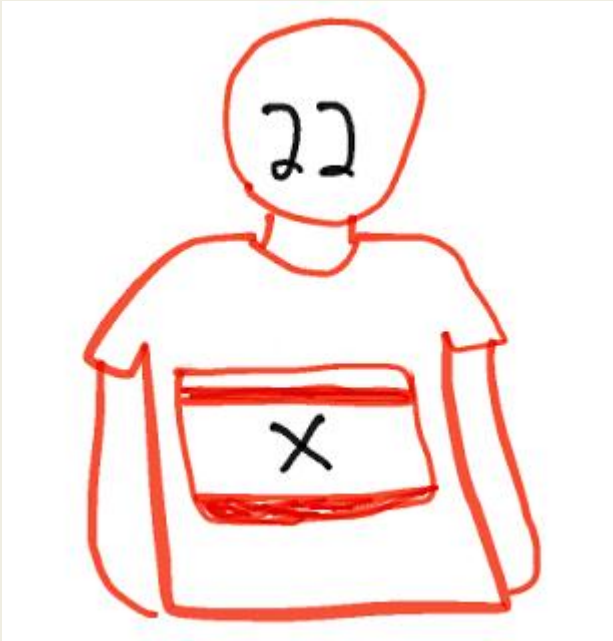
- Functions
  - f(x) = 3x+4
  - A “black box” that accepts input and returns an output
  - Allows you to perform things mu[illegible] instead of copying and pasting.
  - Ex: robot.drive(FORWARD, .5)

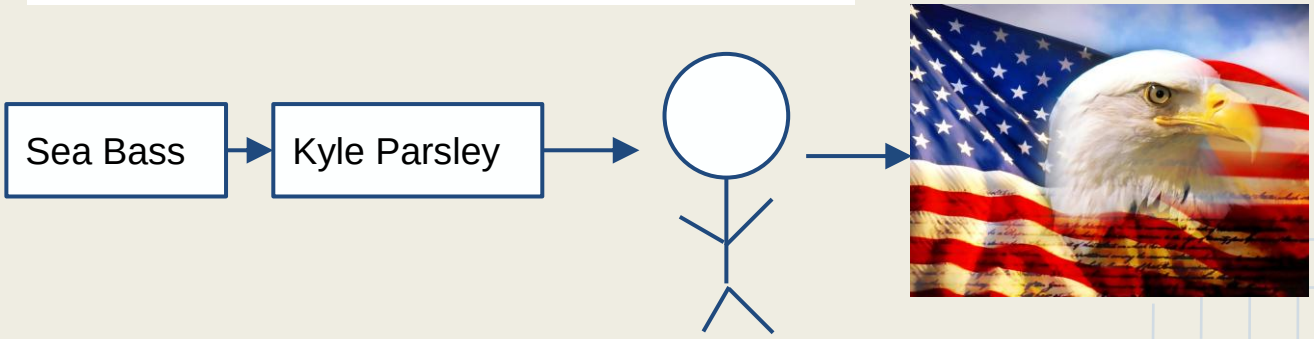




Universal Concepts cont.

- Variables
  - A name represents a value
  - Sea Bass represents Kyle Parsley, a name that represents that guy
    (But that guy also represents a value, an American hero)

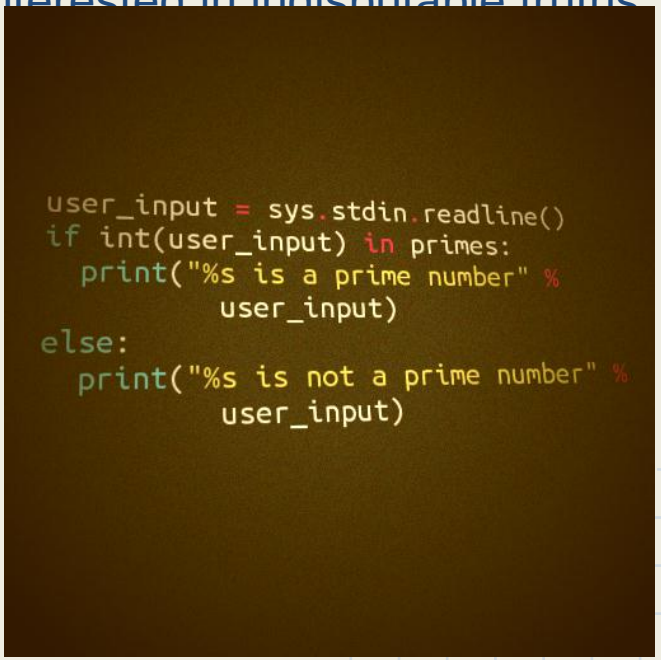




Images courtesy of Khan Academy

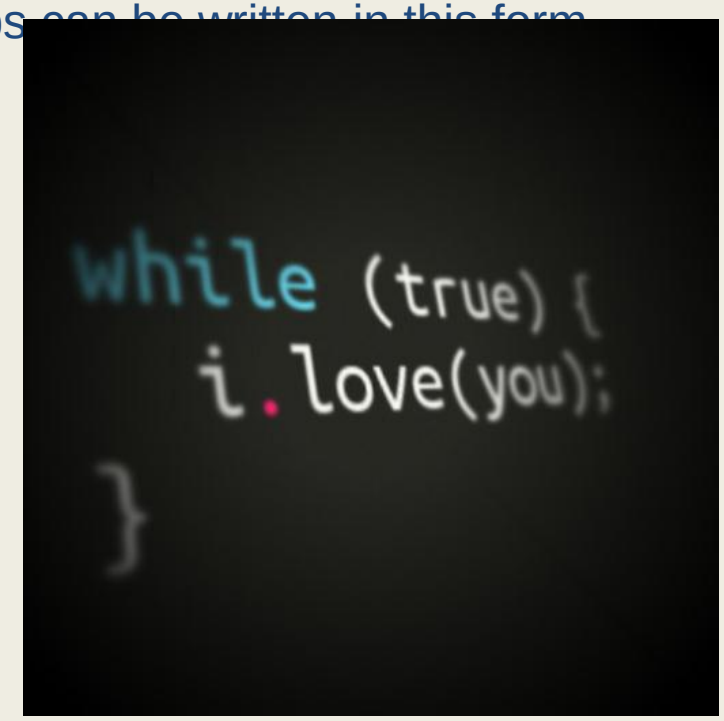
Universal Concepts cont.

- Truthiness
  - Computers are obsessed with the idea of “truthiness”
  - These range from 5 > 3 (an indisputable truth) to Sea Bass thinks he’s a hipster (something with more grey area), or Emacs is better than Sublime Text (more grey areas)
  - However, a computer is only interested in indisputable truths
  - True: Indisputable truth
  - False: Indisputable lie
- If Statements
  - If <statement> do <task>
  - Allows the program to make decisions



Universal Concepts cont.

- Loops
  - Computers excel™ at doing boring, repetitive tasks over and over
  - While <condition> do <task>
    - “Computer, do this task while this condition is truthy”
  - All loops can be written in this form

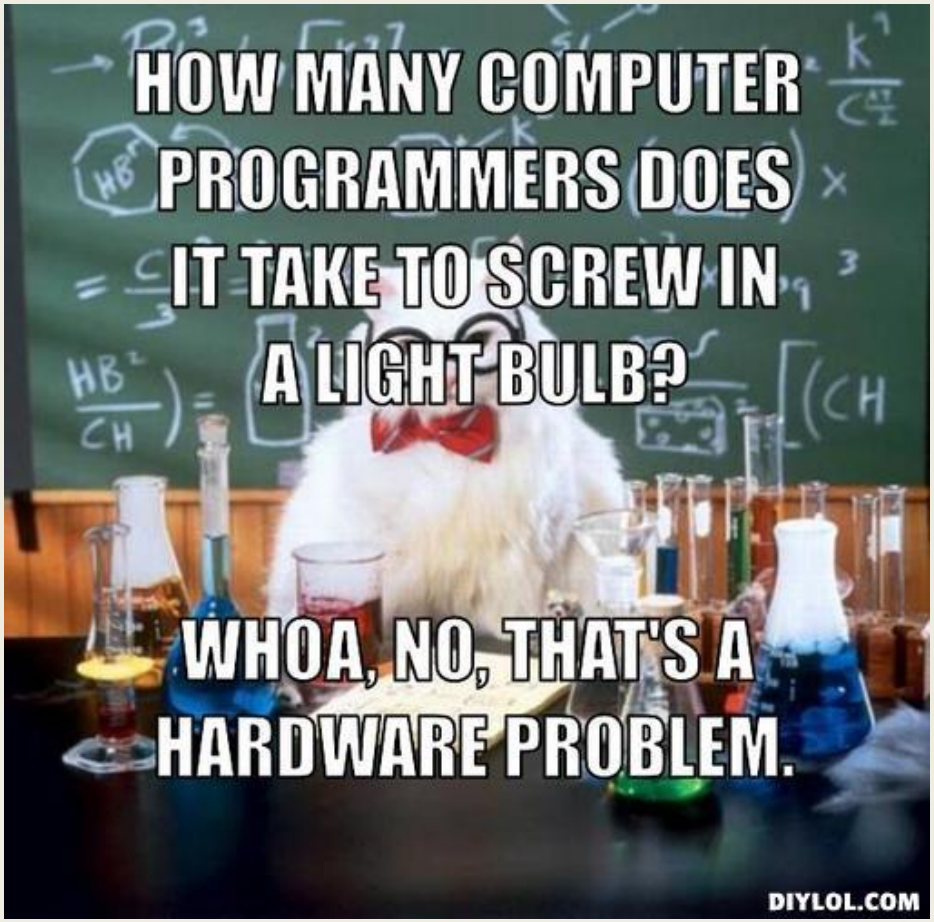


Universal Concepts Cont.

- Don’t Repeat Yourself (DRY)
  - Functions allow you to define blocks of code you can reuse, instead of having to copy and paste it throughout the program.
  - Imagine if every time you wanted move the robot forward, you had to reference every motor, determine a correct power level, pass the digital value along the pipeline to reach the motor
- Keep it Simple, Stupid (KISS)
  - Complex schemes are almost never the right answer (even though they are more fun, typically)
  - Simple systems are easy to maintain
- Garbage In, Garbage Out
  - Never blame the computer, it will execute your commands perfectly
  - If you encounter a problem, it is because of a faulty input

Universal Concepts cont.

- It's *always* hardware ('nuf said)



First Robotics

- What can we do in robotics
- How do we accomplish this

FIRST Robotics

In FIRST Robotics, we typically use C++, Java, or LabVIEW. Each offers its own advantages and disadvantages:

- C++
  - Advantages: Very, very fast. Object-oriented. Other languages. Open source. Lower Level. Fast build times.
  - Disadvantages: Confusing. Syntax. Lower Level.
- Java
  - Advantages: Expressive power. Object-oriented. Other language support. Fast build times. Open source.
  - Disadvantages: Marginally slower. Verbose syntax.
- LabVIEW
  - Advantages: Fast prototyping. Easy to start using. Everything and the kitchen sink. Visual.
  - Disadvantages: Slower build times. Lack of VCS support. Not open source.

Our Club's Language

We use LabVIEW to program the robot, because it is useful for prototyping and testing. Some find a visual representation of code more understandable.

Capabilities and Expectations

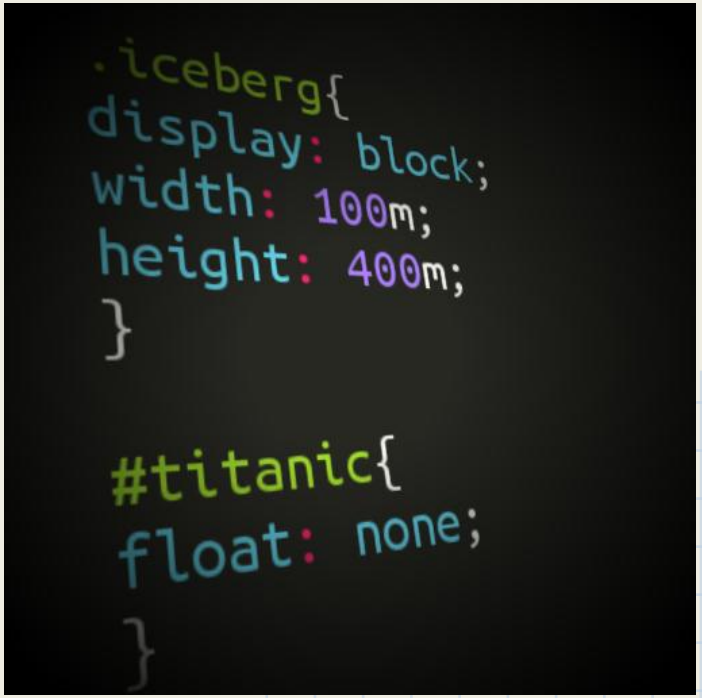
- There are limits as to what programming can accomplish
  - We can’t program the robot to fly (yet)
- How much we can accomplish with programming depends upon many things (in order of most significant to least)
  - The cooperation of the entire team
  - Motivation and time
  - The skill of our programmers
  - Physical limitations (darn you physics)
- Not everyone can be a master at everything
  - Just like every other team, there will be some who do most of the work
  - However, everyone should have basic knowledge about
    - what is feasible or outrageous
    - how to run and debug programs when the main programmers aren’t there

Physics, stahp.

Other FIRST Programming

- The team website
  - We use another language, Python, for the website
  - There's also PHP, Ruby, and other languages

  - HTML defines the data of the website
  - CSS defines how it looks
  - JavaScript defines how it acts



P.S. PHP, Ruby, and most web languages are terrible

It’s over! What next?

- If you’re still interested in programming (and we haven’t bored you to death yet…) then feel free to talk to one of us either in person or by email.
  - Tanner Hobson: thobson125@gmail.com
  - Andrew Wintenberg: awintenb@gmail.com
  - Sea Bass: parsley.kyle@gmail.com
- If you want to learn on your own, Code Academy is a good source for general programming (specifically, JavaScript, HTML, Java, and a lot more). For LabVIEW, there’s some tutorial from many years ago that’s actually alright which we could help you find.