

# APS106



## introduction.

**Week 1 | Lecture 1 (1.1)**

# This Week's Content

- **Lecture 1.1**
  - Introduction
- **Lecture 1.2**
  - Computers, Programs, and The Coding Toolbox
- **Lecture 1.3**
  - Variables, Expressions, and Operators

# Teaching Team



**Ben**  
Instructor



**Seb**  
Instructor



**Joseph**  
TA



**Katia**  
TA



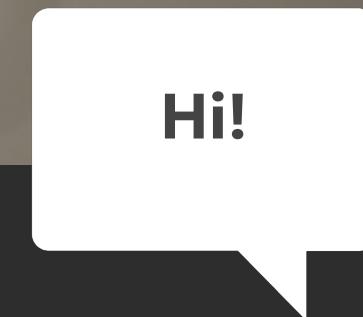
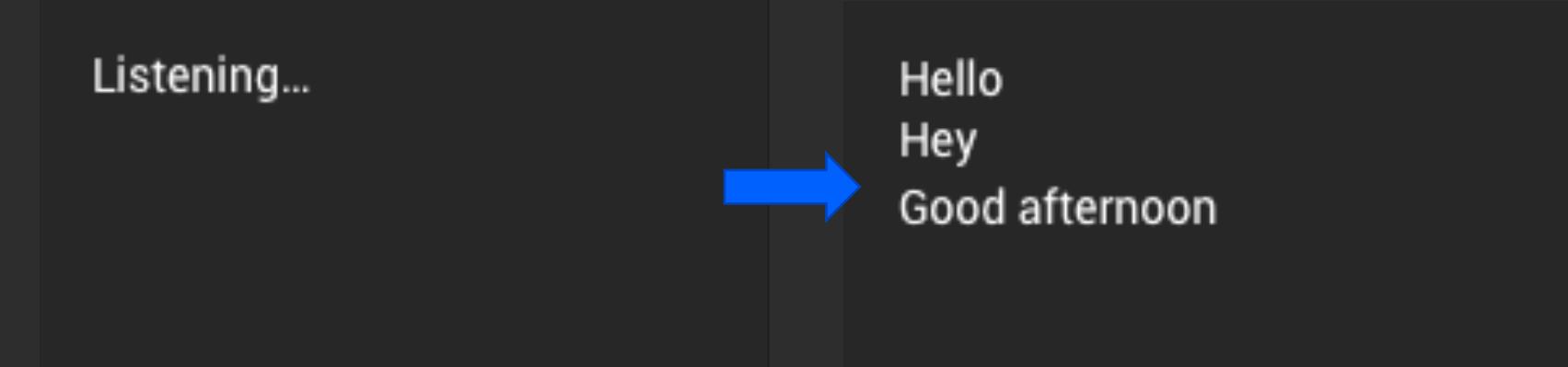
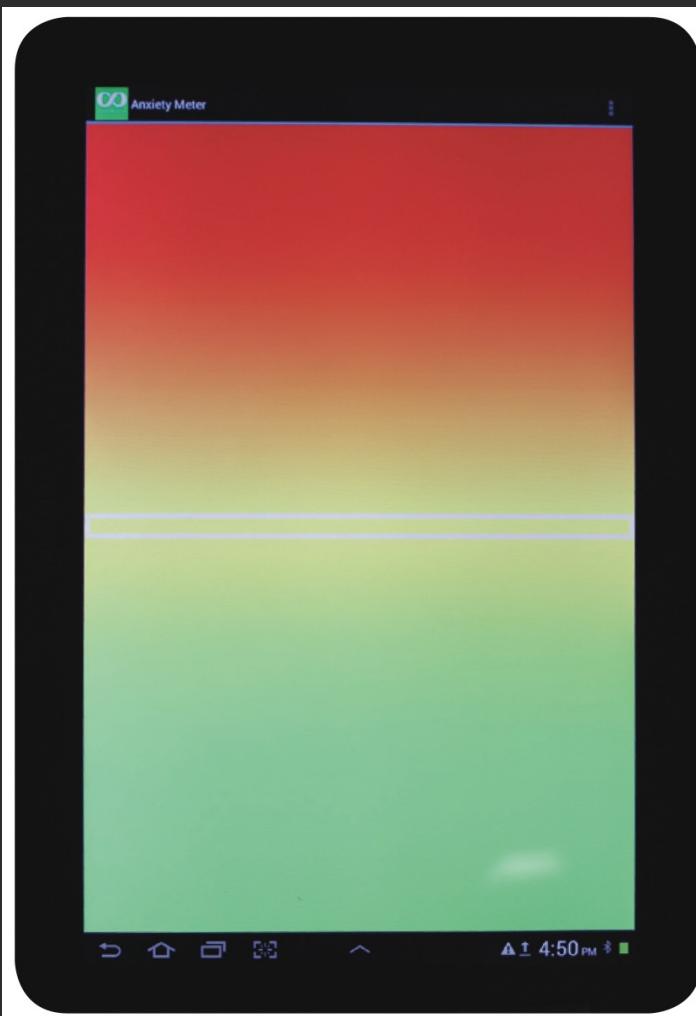
**Sina**  
TA

# Ben

- Bachelor of Electrical & Biomedical Engineering  
@ McMaster University 2015
  - Realized the power of programming (and how it can be used for good, not evil)
- Master of Biomedical Engineering  
@ University of Toronto 2020
  - Hold up... people will pay me for 'playing' on my computer?
- In between obtaining degrees I spent time programming a few different projects...

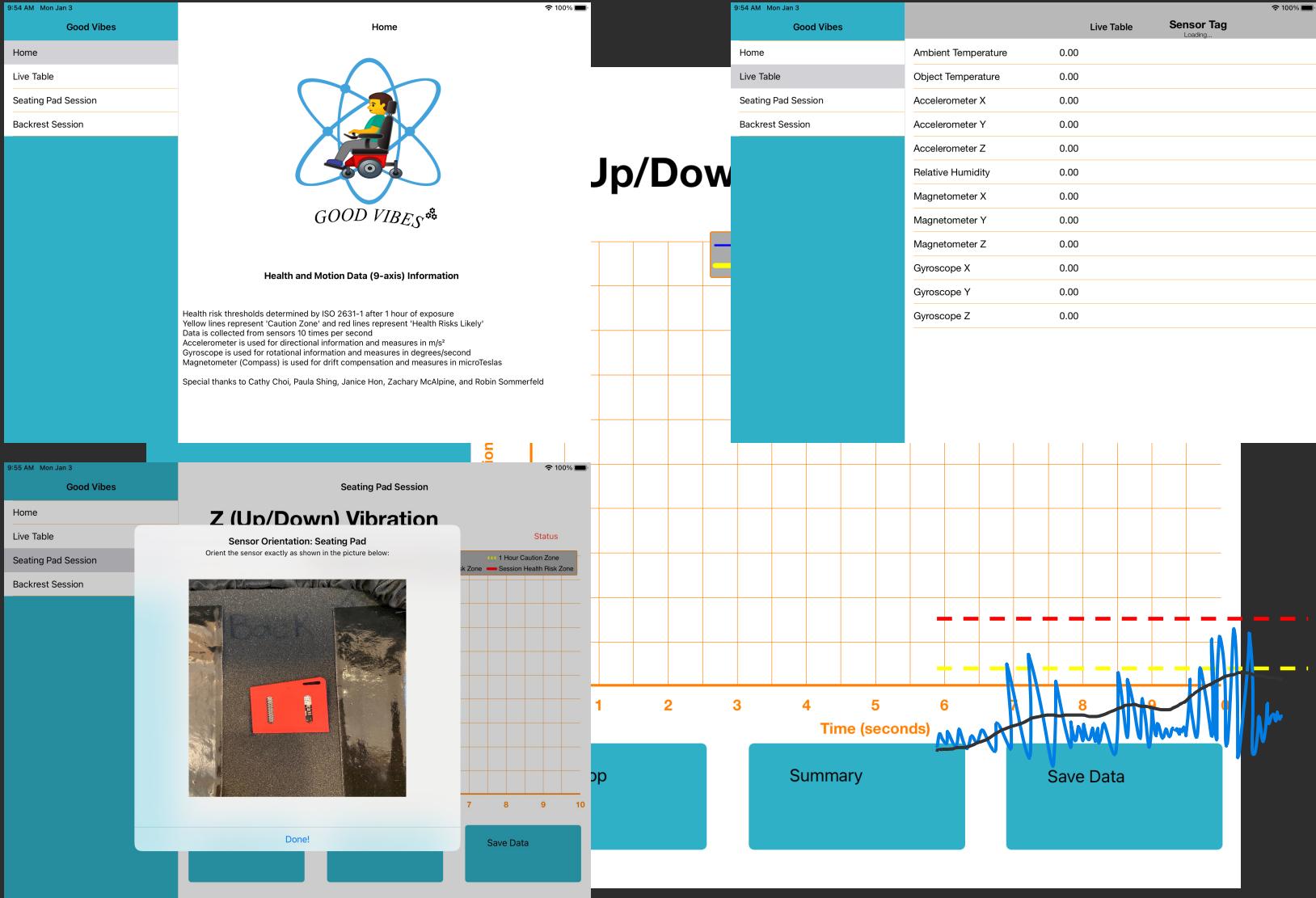


# Projects for children with autism spectrum disorder



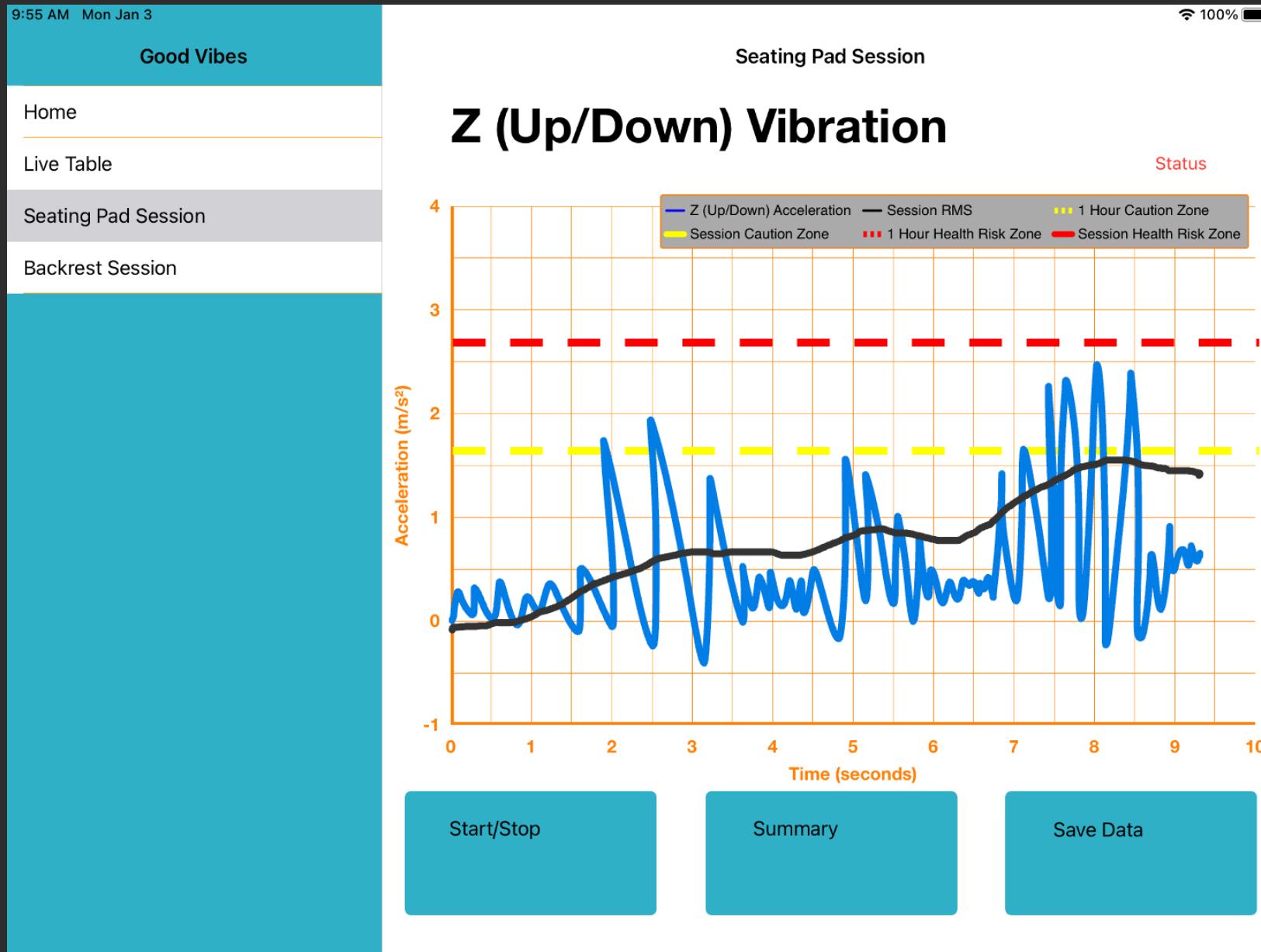
# Good Vibes

## A wheelchair vibration measurement tool



The image displays four screenshots of the Good Vibes mobile application:

- Home Screen:** Shows a sidebar with "Good Vibes", "Home", "Live Table", "Seating Pad Session", and "Backrest Session". The main area features a cartoon character in a wheelchair with a blue atomic-like background and the text "GOOD VIBES".
- Health and Motion Data (9-axis) Information:** A detailed technical page explaining sensor thresholds based on ISO 2631-1, data collection frequency, and sensor types (Accelerometer, Gyroscope, Magnetometer).
- Seating Pad Session Setup:** A camera viewfinder showing a red rectangular sensor placed on a dark surface. Instructions say to "Orient the sensor exactly as shown in the picture below".
- Vibration Analysis Graph:** A line graph titled "Z (Up/Down) Vibration" showing "Time (seconds)" from 1 to 10. It displays a fluctuating blue line with yellow dashed horizontal lines indicating the "1 Hour Caution Zone" and a red dashed horizontal line indicating the "Session Health Risk Zone". Buttons for "Summary" and "Save Data" are at the bottom.



# More about Ben...



# Seb

- Completed CivMin PhD in 2014 where I studies rock fracture and seismology ([RFDF](#)).
- Senior Director of Product & AI at KORE Geosystems, a mining tech startup.
- Senior Research Scientist at SickKids.  [SickKids](#)®
- Joined UofT CivMin in January 2020.
- Research topics: rock mechanics, ultrasonics, signal processing, computer vision, applied machine learning in mining.

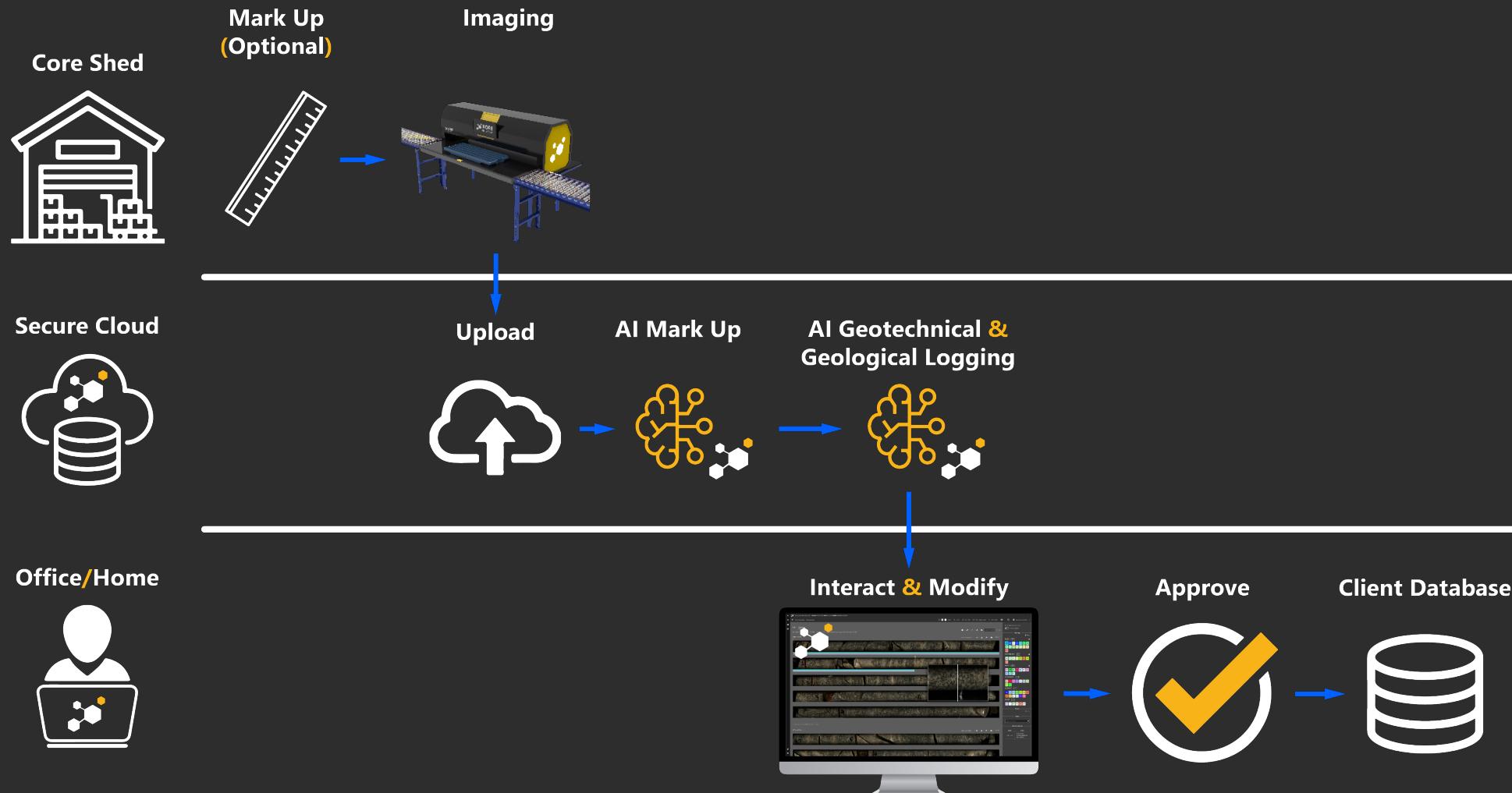


# Seb

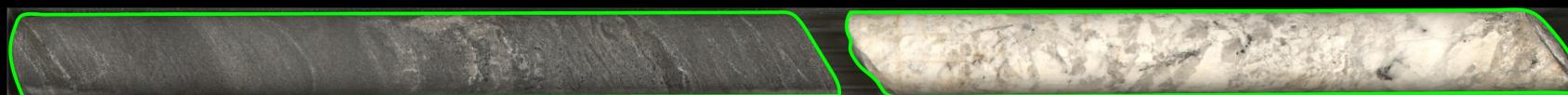
- Completed CivMin PhD in 2014 where I studies rock fracture and seismology ([RFDF](#)).
- Senior Director of Product & AI at KORE Geosystems, a mining tech startup.
- Senior Research Scientist at SickKids.  [SickKids](#)
- Joined UofT CivMin in January 2020.
- Research topics: rock mechanics, ultrasonics, signal processing, computer vision, applied machine learning in mining.



# Building Tech in Mining



# Building Tech in Mining

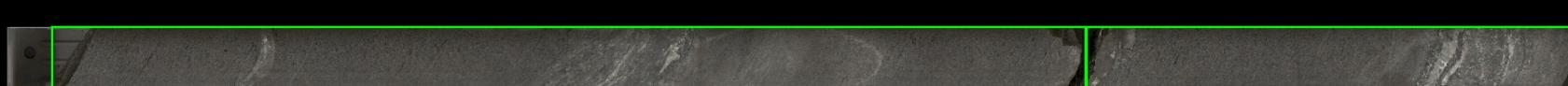
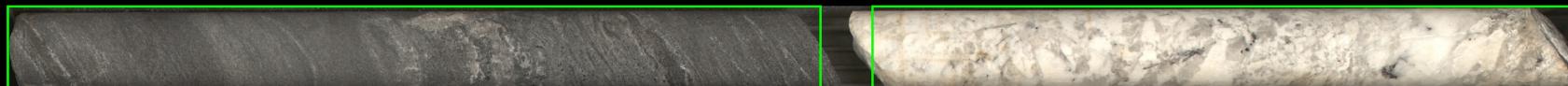


# Applications in Civ & Min

## Fracture Alpha and Beta Angles.



# Building Tech in Mining



# Seb

- Completed CivMin PhD in 2014 where I studies rock fracture and seismology ([RFDF](#)).
- Senior Director of Product & AI at KORE Geosystems, a mining tech startup.
- Senior Research Scientist at SickKids.  **SickKids®**
- Joined UofT CivMin in January 2020.
- Research topics: rock mechanics, ultrasonics, signal processing, computer vision, applied machine learning in mining.



# AI assistance in healthcare.

AI  
Detects  
JET  
Pages  
Mjaye



Staff Physician



JET  
Onset

Nurse



5 minutes later

Treatment  
begins



AI  JET: 5 %

# I like doing other things

- I love travelling and have visited over 20 countries including France, Switzerland, Italy, Austria, Croatia, Costa Rica, Nicaragua, Panama, Peru, Bolivia, United States, Egypt, Turkey, Morocco, South Africa, and Namibia.
- I was a founding member of Ottawa's premiere Beatles cover band.
- I love camping, hiking, and surfing.
- I have a wonderful baby boy named Avery Goodfellow and baby girl name August.
- I love building things with code ( Seb-Good).



# In Person

- Talking
- Working on other courses
- Eating
- Being Late for Lectures



# In Person

- Talking
- Working on other courses
- Eating
- Being Late for Lectures



# Ask questions!

- Ask questions as much as possible.
- Ask each other questions and help if you think you know the answer
- Never forget - if you have a question, there's always other people in the room wondering the same thing. They will be happy you asked it!



# Course Information

- **Quercus:**

- APS106H1 S LEC0101 (Website contains all information).

- **Textbook:**

- ***Programming in Python 3.***
  - A link to the UofT bookstore where you can purchase the textbook can be found on the Syllabus page.

- **Piazza:**

- [piazza.com/utoronto.ca/winter2019/aps106](https://piazza.com/utoronto.ca/winter2019/aps106)



# Grading

Petitions: First-year Office.

Reflections (13 in total)

total: 2%

Labs (10 in total)

total: 10%

Term Test 1 (Feb 6)

total: 15%

Term Test 2 (March 21)

total: 23%

Final Exam

total: 50%

# Lecture Schedule

Section	Monday	Tuesday	Wednesday	Thursday	Friday
LEC0101		4:00-5:00 pm		4:00-5:00 pm	4:00-5:00 pm

- Lectures will be in person in **MYHAL 150**.
- Lecture content can be found on Quercus using the [Kinsella & Goodfellow](#) links.

# Tutorial Schedule

One online tutorial TDB.

Week	Tut Content	Monday	Tuesday	Wednesday	Thursday	Friday
<b>Week 1</b> (Jan 8 - Jan 12)	No Tutorial	No Tutorial	No Tutorial	No Tutorial	No Tutorial	No Tutorial
<b>Week 2</b> (Jan 15 - Jan 19)	<b>Tutorial 1</b>	<b>TUT0104</b> 10:00 AM - 11:00 AM <b>TUT0107</b> 2:00 PM - 3:00 PM	<b>TUT0108</b> 9:00 AM - 10:00 AM <b>TUT0102</b> 3:00 PM - 4:00 PM <b>TUT0105</b> 4:00 PM - 5:00 PM <b>TUT0106</b> 4:00 PM - 5:00 PM	<b>TUT0101</b> 2:00 PM - 3:00 PM <b>TUT0103</b> 2:00 PM - 3:00 PM	No Tutorial	No Tutorial

# Lab Schedule

Two online labs TDB.

Week	Monday	Tuesday	Wednesday	Thursday	Friday
<b>Week 1</b> (Jan 8 - Jan 12)				<b>Lab 0 Released</b> 6:00 PM	<b>PRA0105</b> 3:00 PM - 5:00 PM <b>PRA0106</b> 3:00 PM - 5:00 PM
<b>Week 2</b> (Jan 15 - Jan 19)	<b>PRA0107</b> 9:00 AM - 11:00 AM <b>PRA0103</b> 1:00 PM - 3:00 PM		<b>PRA0102</b> 1:00 PM - 3:00 PM	<b>PRA0101</b> 10:00 AM - 12:00 PM <b>PRA0104</b> 12:00 PM - 2:00 PM <b>PRA0108</b> 12:00 PM – 2:00 PM <b>Lab 1 Released</b> 6:00 PM	<b>PRA0105</b> 3:00 PM - 5:00 PM <b>PRA0106</b> 3:00 PM - 5:00 PM <b>Lab 0 Due</b> 11:59 PM

# Reflections

- There are 12 reflections throughout the term.
- Reflections are worth a total of 2% of your final grade.
- Full marks for successful completion of all reflections.
- Reflections are used to take the pulse of students in the course.
- Insights from reflections are used to iteratively improve the course week-by-week.
- Reflections are there to help you, so please answer honestly.

# Reflections

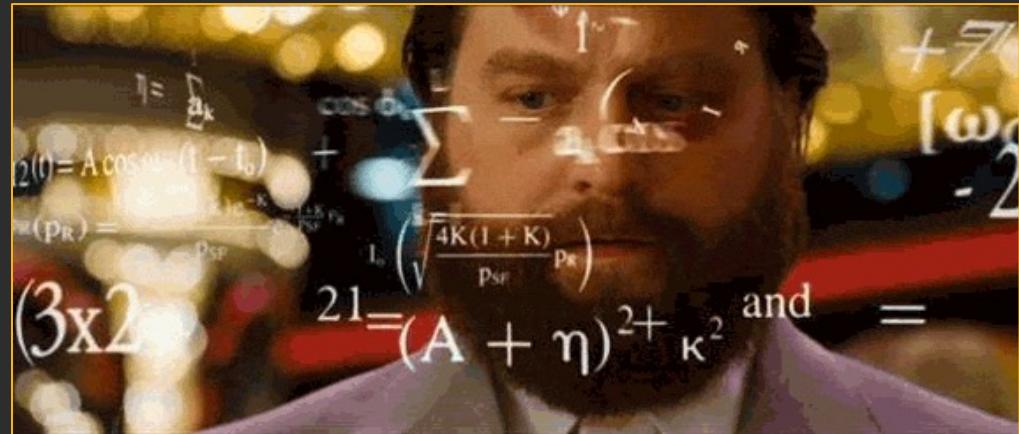
- Reflection include question like:
  - Did you attend lecture this week?
  - Do you understand the topics covered this week?
  - Did you attend tutorial this week?
  - Which lecture section did you attend?
  - What will this code output?
  - Write one word to describe how you feel about the course.

# Reflections Schedule

Week	Reflection	Monday	Tuesday	Wednesday	Thursday	Friday
<b>Week 1</b> (Jan 8 - Jan 12)	<b>Reflection 1</b>				<b>Reflection 1</b> Released 6:00 pm	
<b>Week 2</b> (Jan 15 - Jan 19)	<b>Reflection 2</b>	<b>Reflection 1</b> Due 1:00 am			<b>Reflection 2</b> Released 6:00 pm	
<b>Week 3</b> (Jan 22 - Jan 26)	<b>Reflection 3</b>	<b>Reflection 2</b> Due 1:00 am			<b>Reflection 3</b> Released 6:00 pm	
<b>Week 3</b> (Jan 29 – Feb 2)	<b>Reflection 4</b>	<b>Reflection 3</b> Due 1:00 am			<b>Reflection 4</b> Released 6:00 pm	

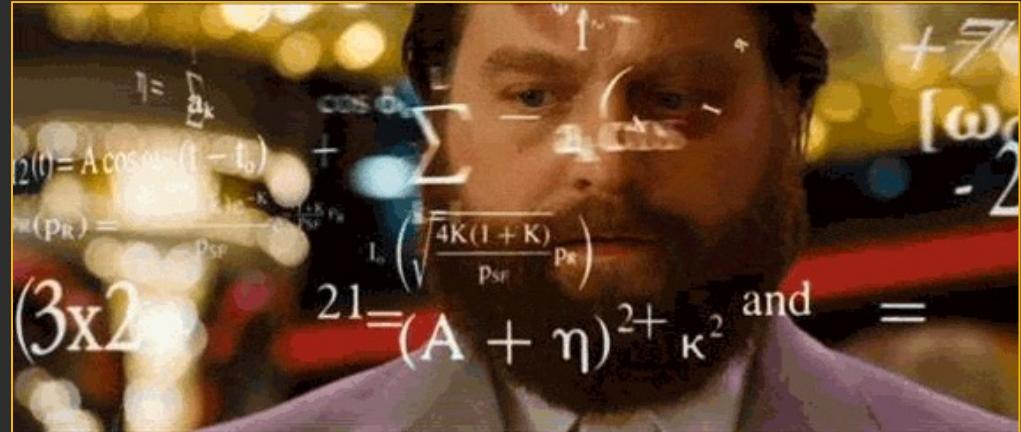
# Tips for Success

- Do not wait until the last minute.
- Work Together (except during the test/exam):
  - Synthesis
  - Application
  - Evaluation
  - Explanation
- Work Smart (effort doesn't necessarily equal progress).
- Time Management.
- Know when (and how) to ask for help.



# Tips for Success

- **88%** of your final grade is based on two term tests and an exam.
- For these tests and exams, you'll be to write code.
- You will not be able to achieve a high mark by reading the textbook and lecture note. You need to **PRACTICE, PRACTICE, PRACTICE!**
- How can you practice writing code?
  - **Lecture Breakout Sessions.**
  - **Labs (this of this as studying).**
  - **Practice Problems.**



Writing A Lot Of Code During The Term

=

Higher Grade

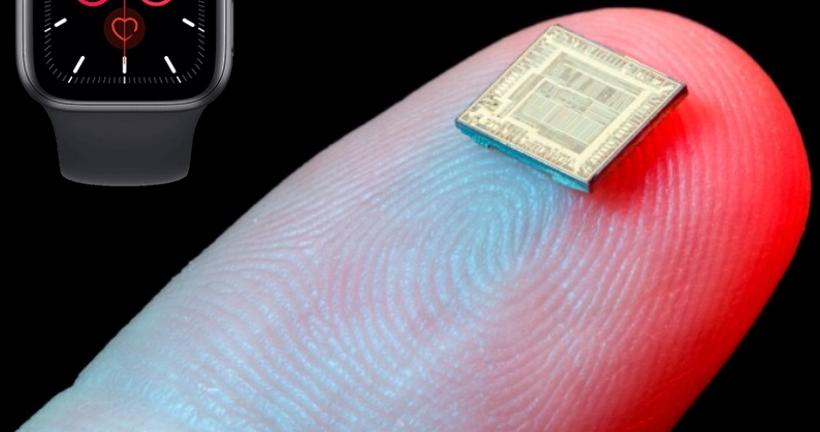
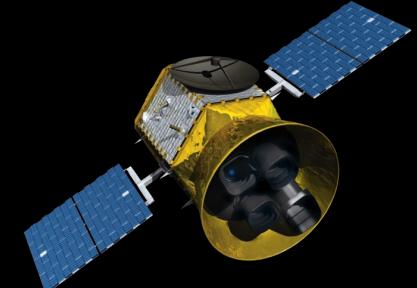
```
31 def __init__(self, path=None, debug=False):
32     self.file = None
33     self.fingerprints = set()
34     self.logduplicates = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, 'seen_requests'), 'a')
39         self.file.seek(0)
40         self.fingerprints.update([line.strip() for line in self.file])
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('SUPERVISOR_DEBUG')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

# What is this course all about?

- Learn the **fundamentals** of programming.
- **Programming as a tool** used in engineering.
- **Engineering design** through programming.

# Why Programming?

StartUp **HERE**



# Why Programming?

- Computers are everywhere:
  - Learn to interact with them.
  - Simplifies our jobs.
- Entrepreneurial opportunities (\$\$\$).
- Necessary for engineering jobs.
- Research and development.
- Because it's fun!



#DISRUPTMINING

PRESENTED BY EGOLDCORP

CO-HOSTED BY KPMG

 **KORE**  
GEOSYSTEMS

\$1,000,000 Investment.

Owner of **McEwen  
Mining Inc.**CEO of **Franco-  
Nevada Corporation**President of **Cisco  
Systems Canada**COO of **Goldcorp**SHARK on ABC's  
**Shark Tank**

# What is Programming?

- A way of telling a computer what to do.
- A computer can't infer (...yet).
  - Need to tell a computer every single step it needs to do in a language it can understand.
  - How would you request an egg for breakfast to a chef and to a computer/robot?
- **To a Chef**
  1. Sunny-side up, please!
- **To a Computer**
  1. "Turn on stove"
  2. "Take out pan"
  3. "Take one egg out of fridge"
  4. "Crack egg"
  5. "Pour egg into pan"
  6. "Wait 5 minutes"

# How to Program a Computer.

Programmer



English

Pseudocode

Programming Language

Compiler

Assembly Code

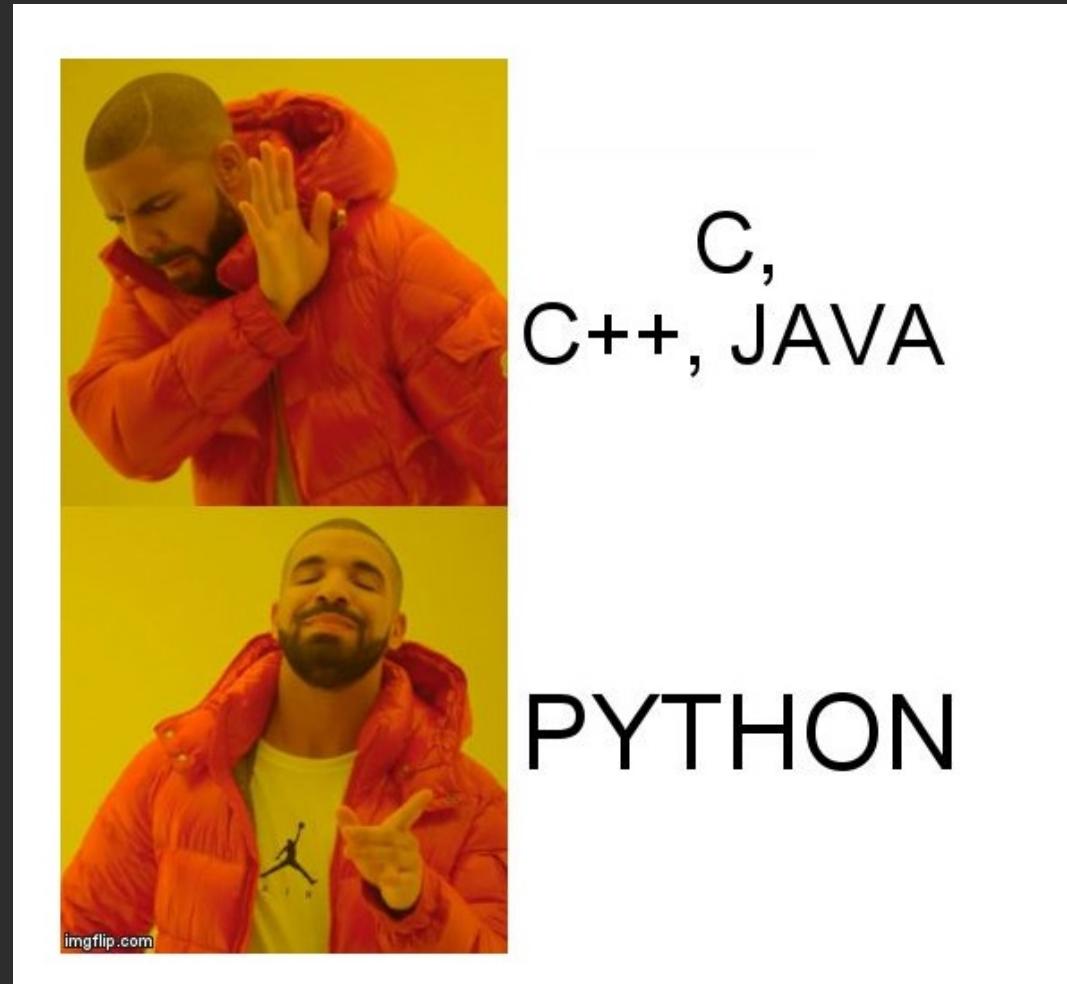
Assembler

Machine Code



 Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



 Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



```
public class HelloWorld
{
    public static void main(String[ ] args)
    {
        system.out.println("Hello World!");
    }
}
```



```
print("Hello World!")
```

Lower-Level

Higher-Level



# Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

Increasing  
Program Size

Matlab, SQL

C++, Java, Python

C, Fortran, Pascal

Assembly Code

Machine Code



Increasing  
Ease-of-Use

 Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



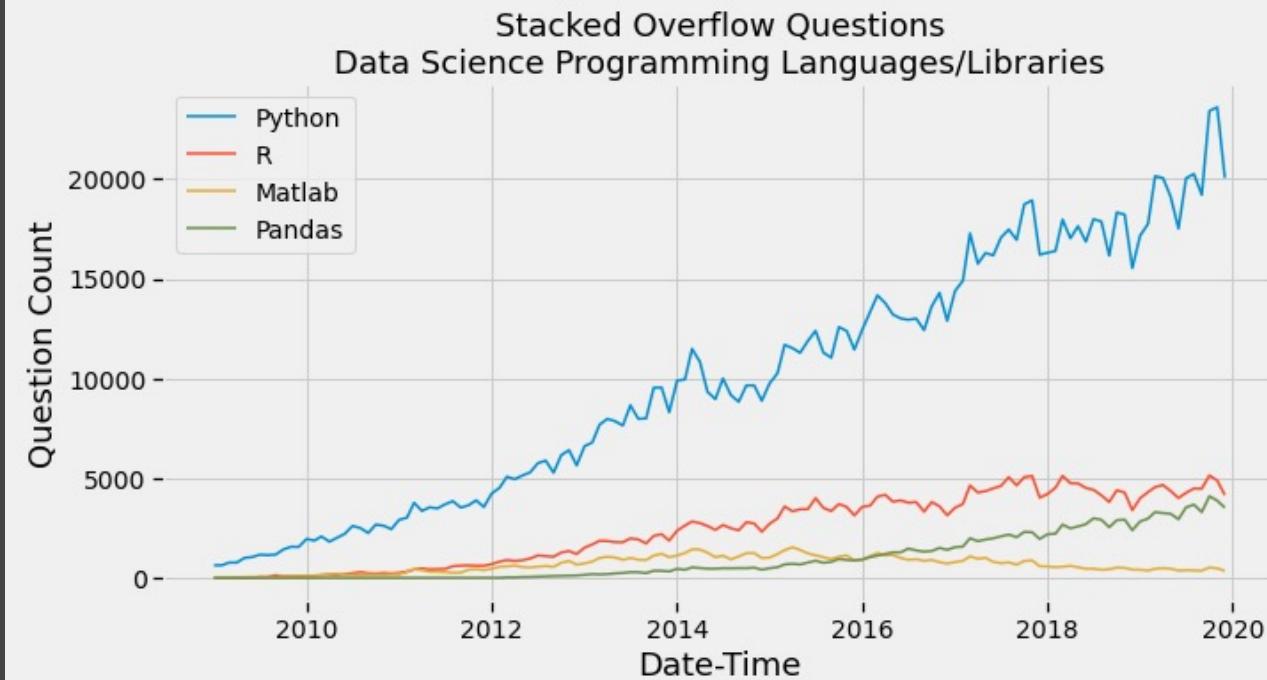
```
if x > 10:  
    print("x is greater than 10")
```



You don't know Python.  
Can you figure out what this code does?

 Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



 Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



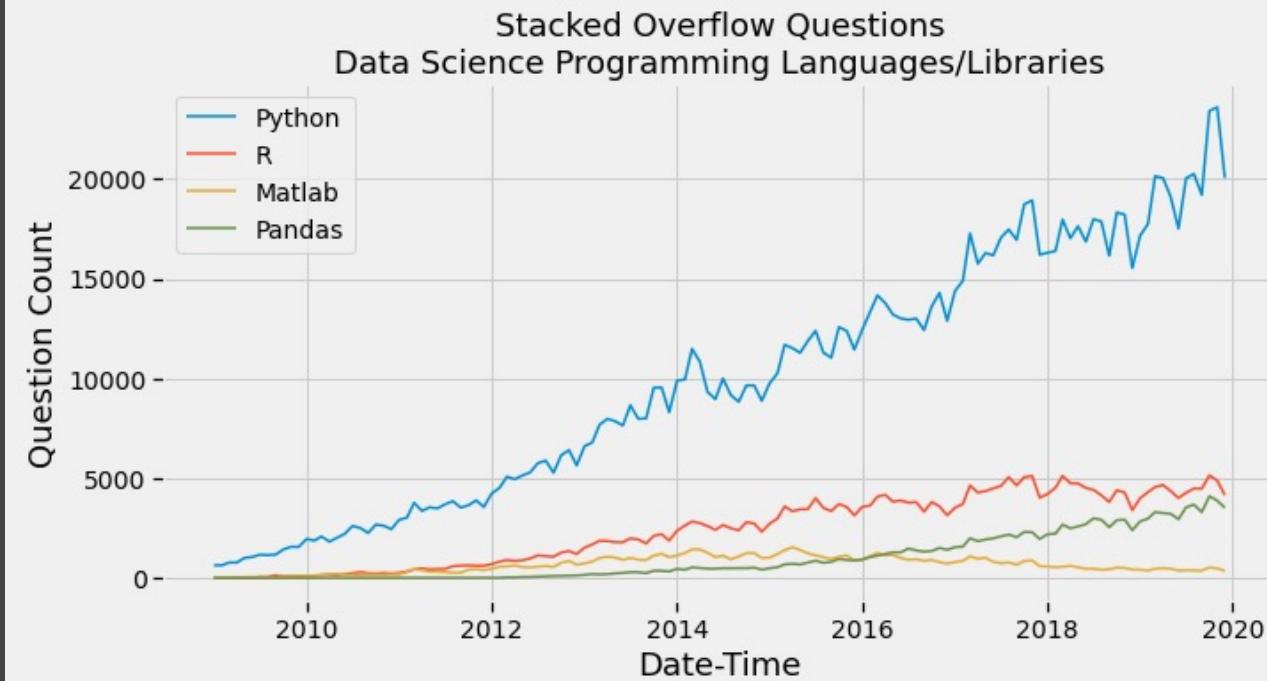
 Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.

 PyTorch LANDING AI Azure  
Machine Learning XGBoost scikit  
learn OpenAI TensorFlow

 Why Python?

- High-level language making it more like a readable, human language than other low-level languages.
- Simple and clear syntax.
- Large open-source community.
- Used by Google, Firefox, Dropbox, Youtube, Instagram, Yelp, NASA, CIA, etc.
- The Artificial Intelligence (AI) community overwhelmingly uses Python.



# Why Engineers?

- Ok, so we've explained why coding is important and why we're learning to code in Python.
- **BUT**, you're all engineers, right?
  - Mechanical
  - Materials
  - Civil
  - Chemical
  - Mineral
- Why do these kinds of engineers need to know how to code?



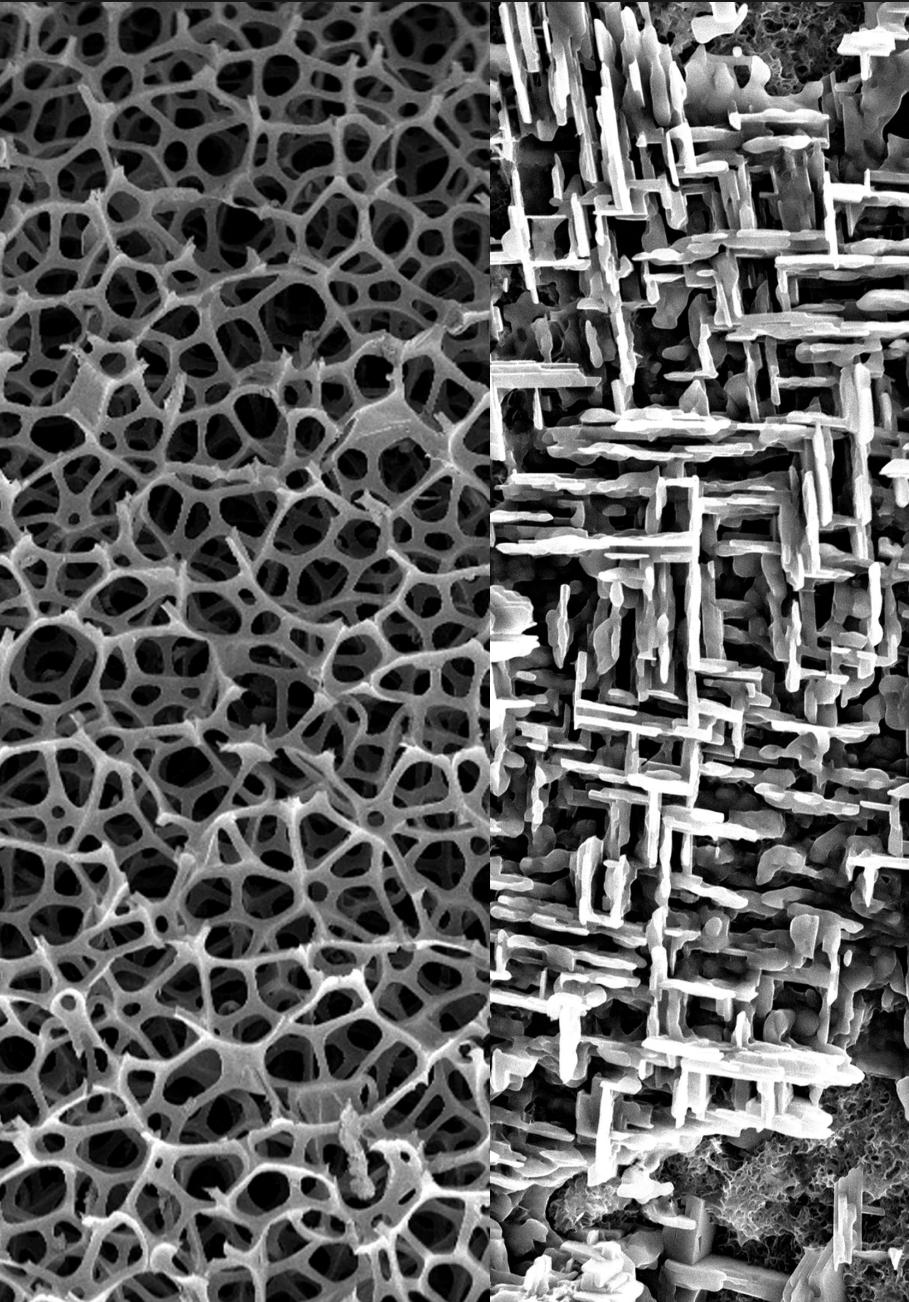
# Mechanical Engineers

- Design Airplanes
- Micro Fluid Systems
- Automotive Engines
- Hydraulic Dams



# Materials Engineers

- Biomaterials
- Design of Materials
- Manufacturing with Materials
- Sustainable Materials Processing



# Civil Engineers

- Design Buildings
- Test Concrete
- Manage Water Supply System
- Design Bridges



# Chemical Engineers

- Chemical Manufacturing Processes
- Petroleum Extraction
- Nanotechnology
- Manufacturing Computer Components



# Mineral Engineers

- Mine Design
- Mineral Processing
- Mineral Exploration
- Mining Finance
- Mineral Extraction



# Why Engineers?

- Again, why do engineers need to know how to code?
- In the 20<sup>th</sup> century, engineers didn't need to know how to code.
- In the 21<sup>th</sup> century, coding is an **essential skill** and any engineer without solid coding abilities will be at a major disadvantage.

```
31      self.file = None
32      self.fingerprints = set()
33      self.logduplicates = True
34      self.debug = debug
35      self.logger = logging.getLogger()
36
37      if path:
38          self.file = open(os.path.expanduser(path))
39          self.file.seek(0)
40          self.fingerprints.update(self._load_fingerprints())
41
42  @classmethod
43  def from_settings(cls, settings):
44      debug = settings.getboolean("debug")
45      return cls(job_dir=settings["job_dir"], debug=debug)
46
47  def request_seen(self, request):
48      fp = self.request_fingerprint(request)
49      if fp in self.fingerprints:
50          return True
51      self.fingerprints.add(fp)
52      if self.file:
53          self.file.write(fp + "\n")
54
55  def request_fingerprint(self, request):
56      return request_fingerprint(self, request)
```

# Why Engineers?

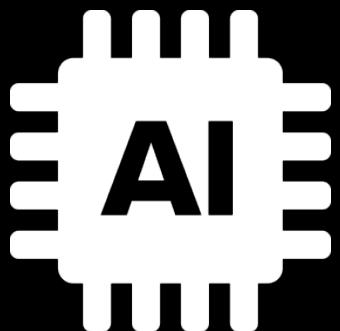
- Five technology trends have changed the game for engineers.



The Cloud



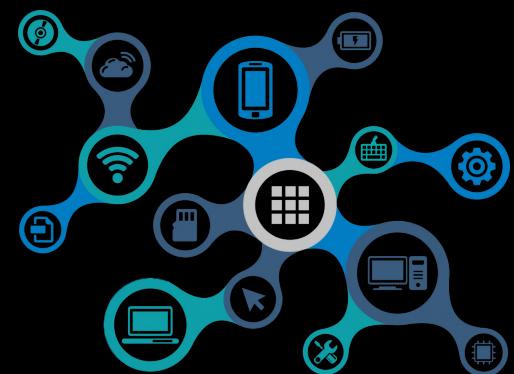
Cheap Sensors



Artificial  
Intelligence



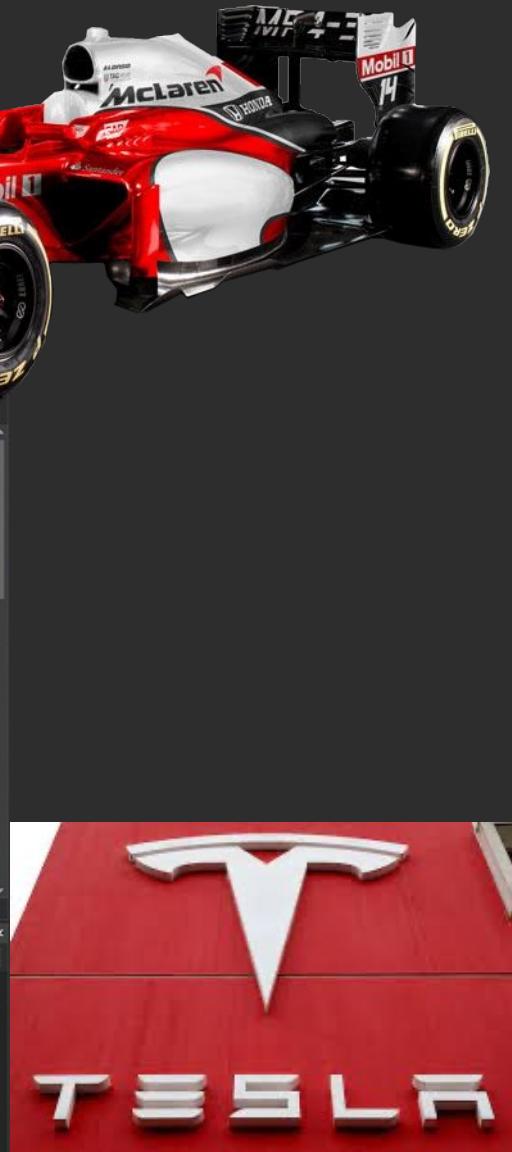
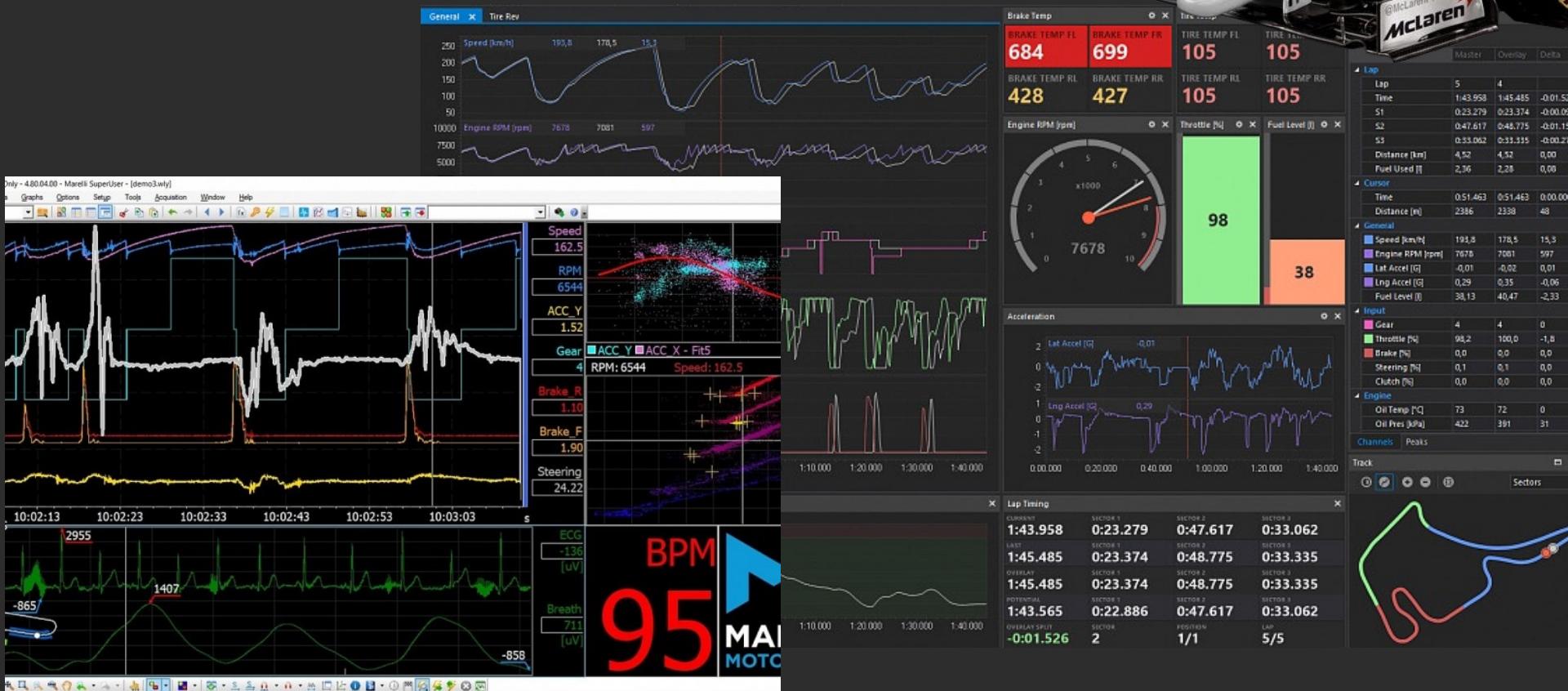
Big Data



The Internet  
Of Things

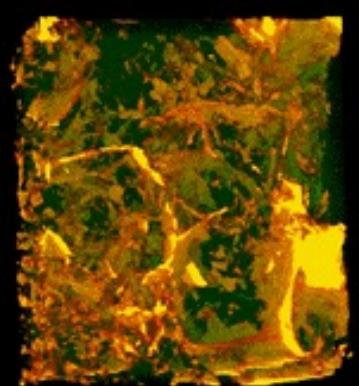
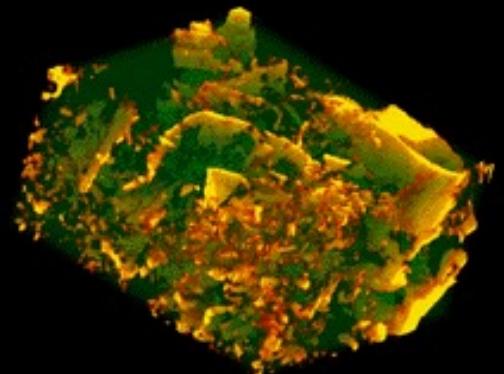
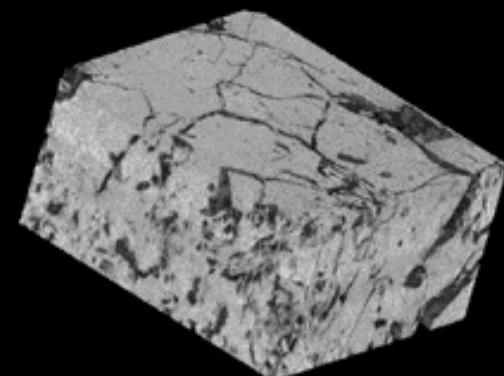
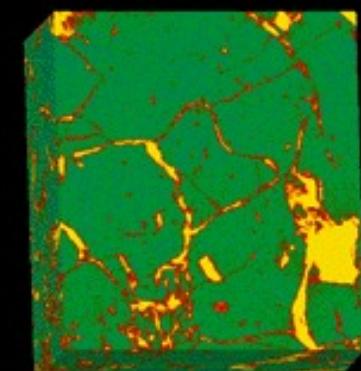
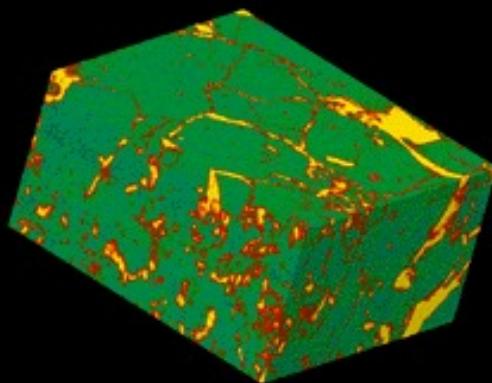
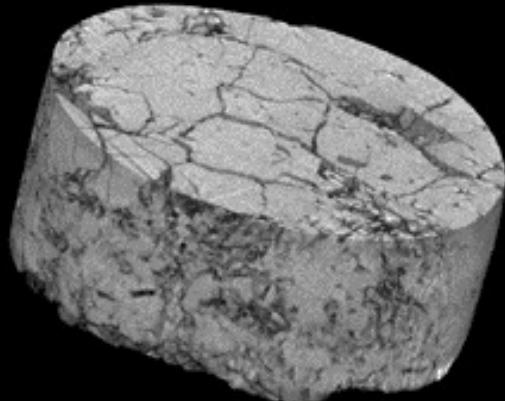
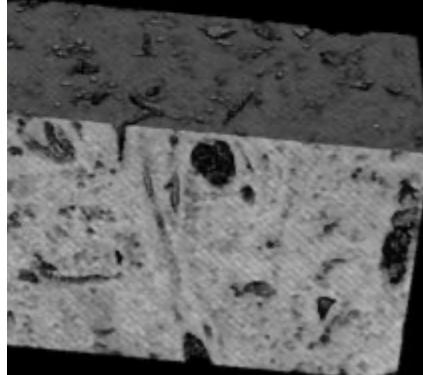
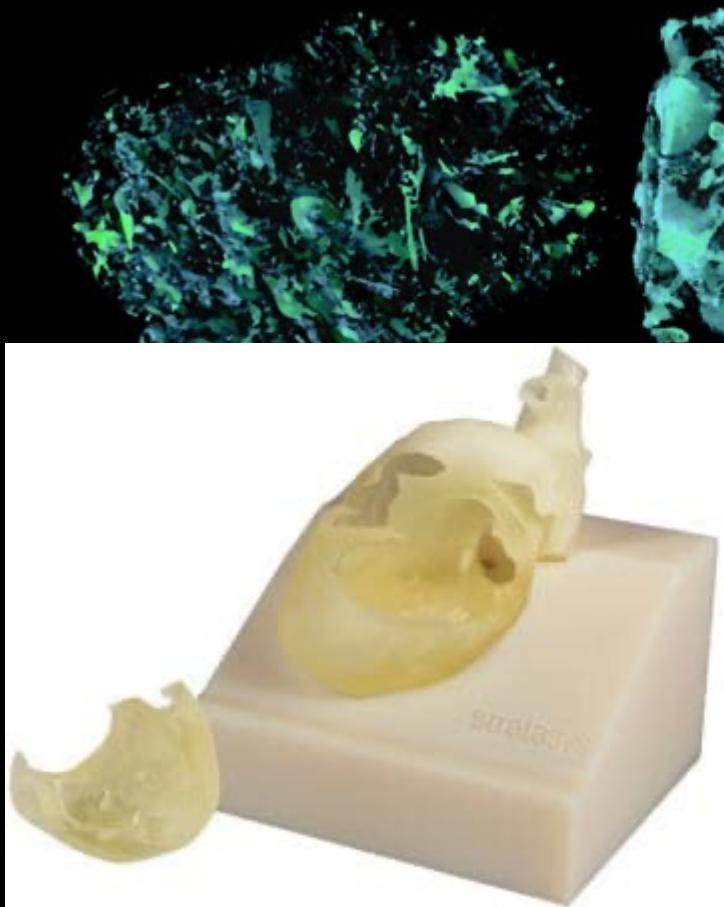
# Mechanical Engineers

## Realtime Diagnostics & Tuning



# Materials Engineers

Digital Image Analysis of Material Microstructure



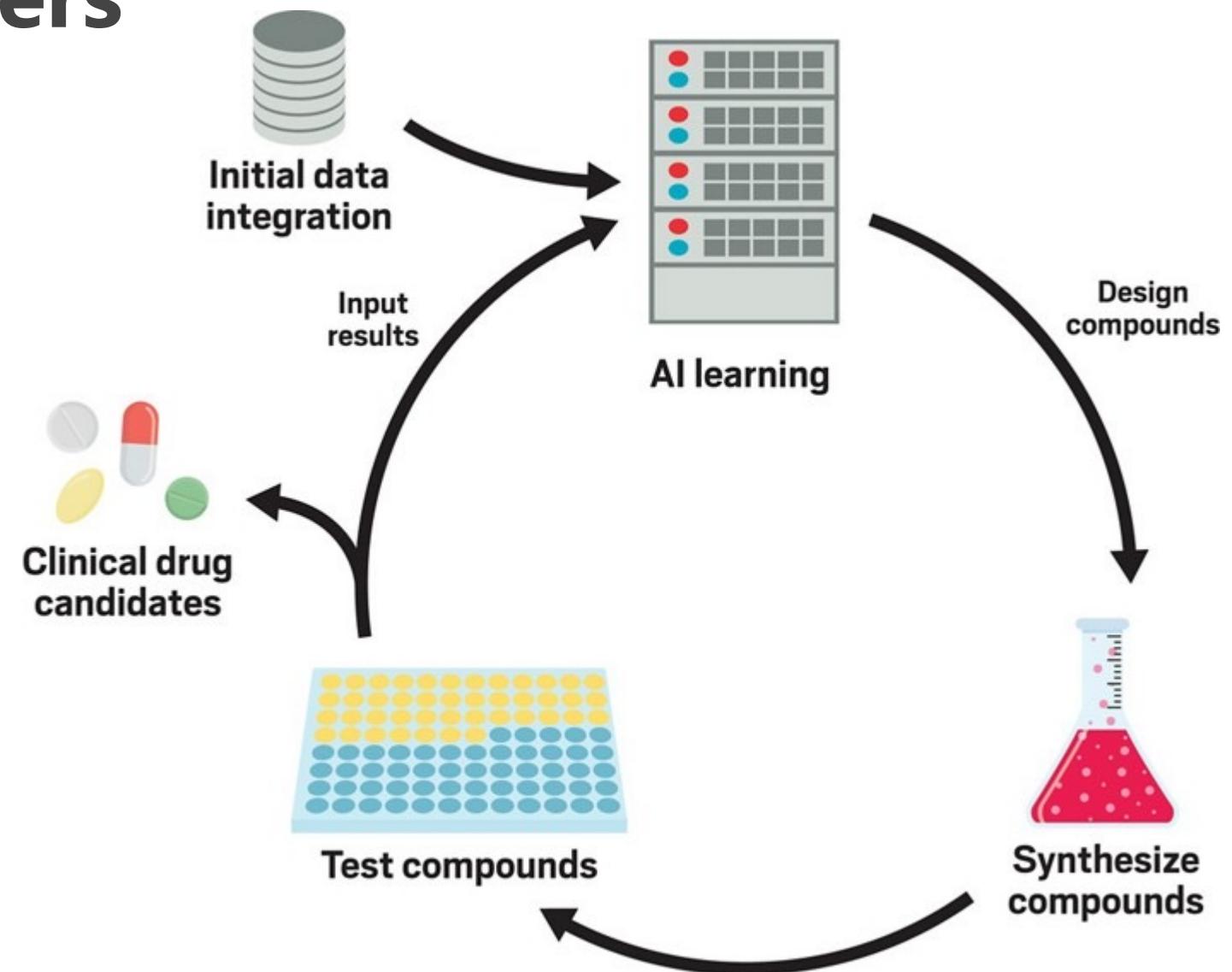
# Civil Engineers

Urban Transportation Disrupted

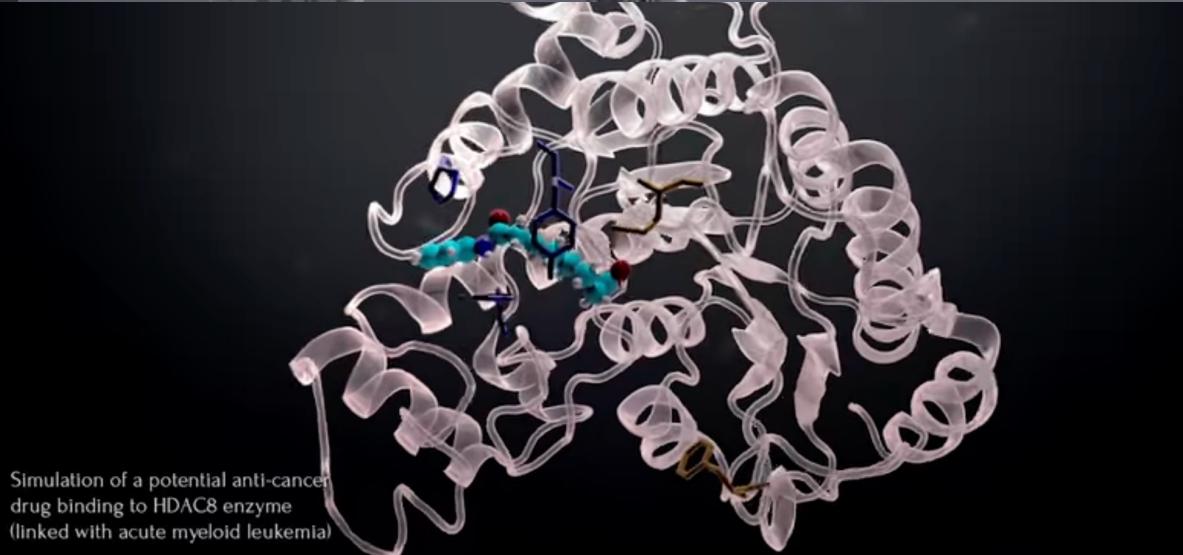


# Chemical Engineers

Synthesizing new compounds



# Sci-fi Inspiration



# Mineral Engineers

Semi-Autonomous Mines



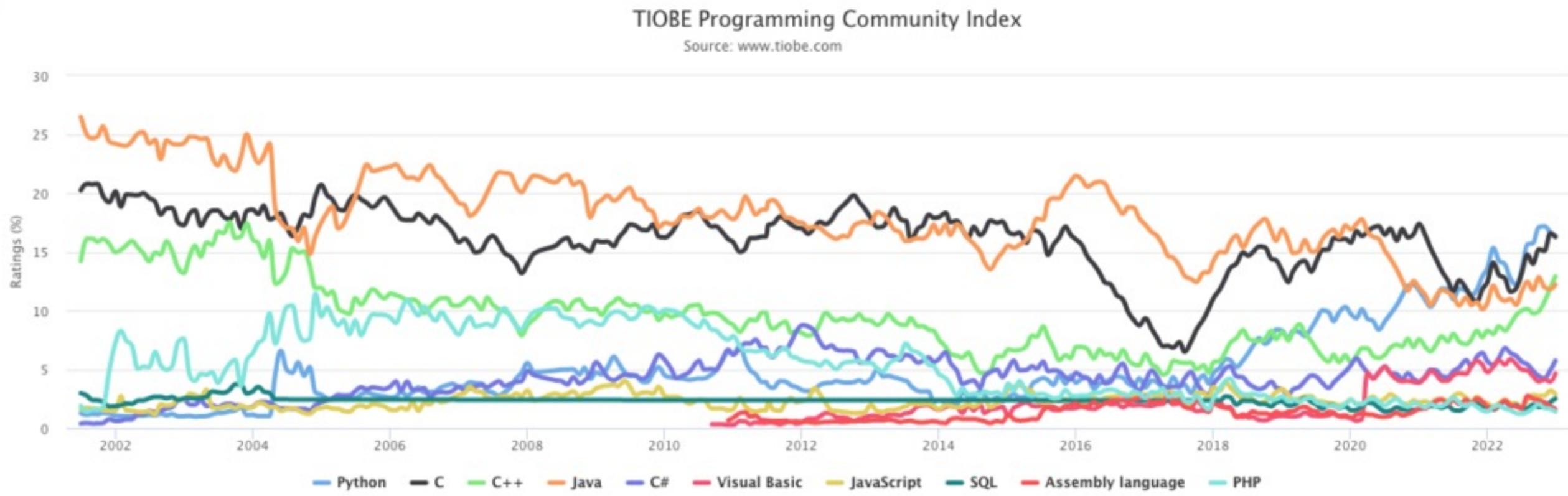
# Programming Skills Are In Demand

- According to the *Developer Survey by StackOverflow*, Python was one of the most in-demand technologies of 2018, 2019, 2020, and 2021.
- As of 2021, it is ranked as the world's 2nd most popular programming language among professional software developers as well as the first most wanted programming language.

```
31     dev = None
32     self.file = None
33     self.fingerprints = set()
34     self.logduplicates = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.expanduser(path))
39         self.file.seek(0)
40         self.fingerprints.update(self._read_file())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getboolean('debug')
45         return cls(job_dir=settings['job_dir'],
46                    request_fingerprint=settings['request_fingerprint'],
47                    request_seen=settings['request_seen'],
48                    logduplicates=settings['logduplicates'],
49                    file=settings['file'],
50                    fingerprints=settings['fingerprints'])
51
52     def _read_file(self):
53         fp = self.file
54         if fp:
55             fp.seek(0)
56             return fp.read()
57
58     def request_seen(self, request):
59         fp = self.request_fingerprint(request)
60         if fp in self.fingerprints:
61             return True
62         self.fingerprints.add(fp)
63         if self.file:
64             self.file.write(fp)
65
66     def request_fingerprint(self, request):
67         return request_fingerprint(self, request)
```

# Python Skills Are In Demand

- According to the *TIOBE Index*, Python overtook C and Java in 2022 as the world's most popular programming language



# One of the reasons for choosing Engineering!

- Finally! A course that isn't Chem, Bio, Physics, or Math
- This (as well as your engineering design course) is what separates you from many other sciences
  - And makes you **MUCH** more valuable....

Programmer



What my friends think I do



What my mom thinks I do



What society thinks I do



What my boss thinks I do



What I think I do



What I actually do

# APS106



## introduction.

**Week 1 | Lecture 1 (1.1)**