Future EECS Courses

Some of the many courses offered here...

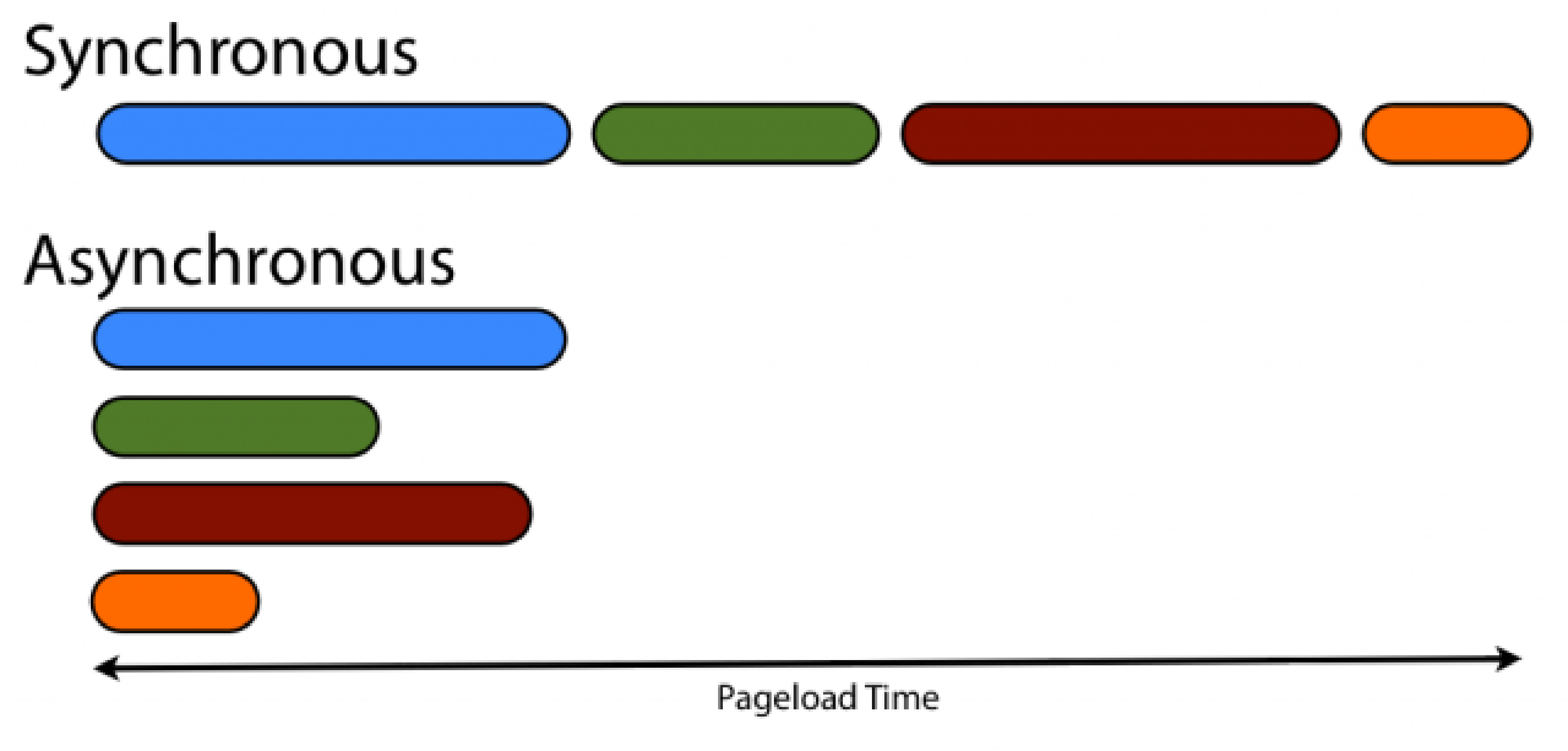
- Creating large software systems
  - EECS 485: Web Systems
  - EECS 388: Computer Security
  - EECS 494: Game Development (MDE/Capstone)
  - EECS 484: Databases
- Getting useful information from data
  - EECS 445: Machine Learning
  - EECS 442: Computer Vision
- Low-level programming concepts
  - EECS 482: Operating Systems
  - EECS 489: Networks
  - EECS 381: Object-Oriented Programming
- Math very relevant to computer science
  - EECS 475: Cryptography
  - EECS 477: Algorithms

- EECS 398: Computing for Computer Scientists (Soon to be EECS 201)
- EECS 490: Programming Languages
- Hardware classes
- Many others

EECS 485: Web Systems

- Learn Python HTML/CSS/Javascript!
- Thinking about asynchronous programming.
- Dealing with large amounts of data.
- Building and using REST APIs (common in industry!)
- Have DeOrio tell you about dumb terminal stuff that we all love (cowsay).
- Learn to automate processes in your workflow.
- With prior web development experience, workload is significantly less than 281.
- Without prior experience, the learning curve for Python and Javascript can be steep.
- 5 projects, 2 exams.

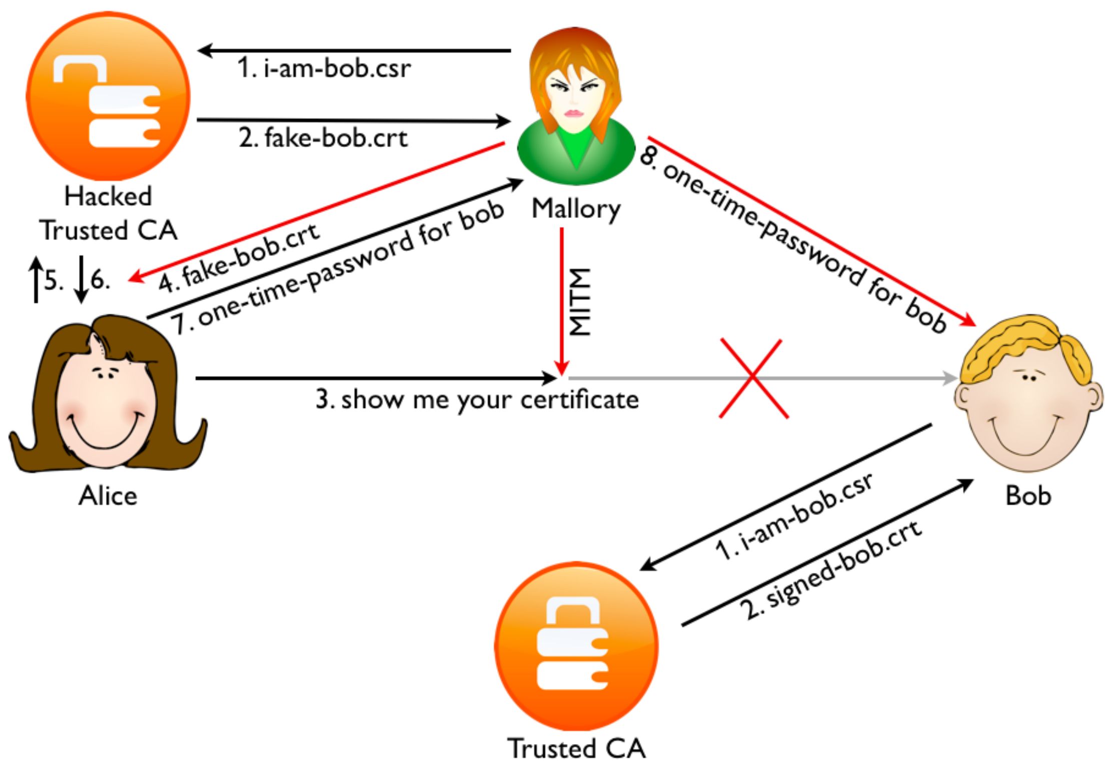
EECS 485: Web Systems



EECS 388: Computer Security

- Project topics: cryptography, network and web security, application security, and digital forensics.
- Learn about vulnerabilities in a wide array of software applications.
  - Mostly web and network security.
- Learn about hashing and why it's extremely important to have collision-resistant hash functions.
- Listen to Alex Halderman speak about voting stuff!
- Workload significantly less than 281.
- 5 projects, 5 homeworks, 1 exam.

EECS 388: Computer Security



The man-in-the-middle attack is a popular topic in 388.

The trick is figuring out how to pass information in a public space so that only the person you intended to pass the information to can understand it.

EECS 494: Game Development

- Learn how video games are built from the ground up.
- Use Unity and C# to create a game of your own.
- Work with a team to consider how a full-scale video game is implemented (and how much time it takes).
- Workload is significantly heavier than 281.

EECS 494: Game Development

Grand Theft Auto 5 is the third best-selling video game of all time, and took (at peak) 1000 engineers nearly five years and $137 million in development costs to create.

EECS 484: Databases

- Learn the foundations of most data management systems in existence today, in terms of both how to use them effectively and how to implement them.
- Learn SQL (Structured Query Language) and Java, continue to use C++, which is widespread in the implementation of database management systems.
- Workload is significantly less than 281.
  - 4 projects, 4 homeworks, 2 exams.
- Project topics:
  - Design and implement a database for facebook data
  - Writing complex SQL queries for the database
  - Implementing a type of tree used for organizing information (AVL tree on steroids).

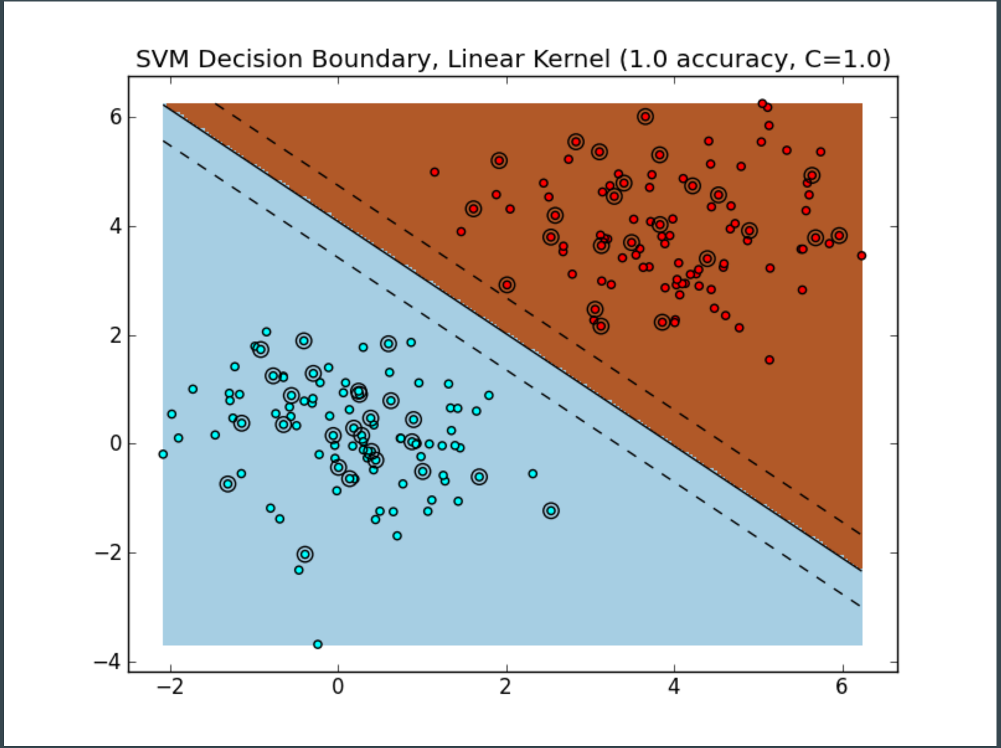
EECS 484: Databases

According to one survey, nearly 79% of applications use SQL databases to store information.

EECS 445: Machine Learning

- Use Linear Algebra, Statistics, and Calculus all at once to solve interesting problems!
- Learn more about Python.
- Understand how we can use large amounts of data to accurately predict or classify actions in the future.
- Learn concepts such as Support Vector Machines, Neural Networks, and Data Clustering to solve classification and regression problems.
- Workload comparable with 281, be ready for math problems (MATH 217/419, 215, and STATS 412/425 good prep - you may struggle without these).
- 4 homeworks, 2 projects, 2 exams.
- Word of caution: lectures are NOT recorded

EECS 445: Machine Learning



Almost all of modern machine learning (including neural networks) is about finding a line that separates your data into meaningful segments.

The trick is to get a computer to do it, and attempt to do it efficiently.

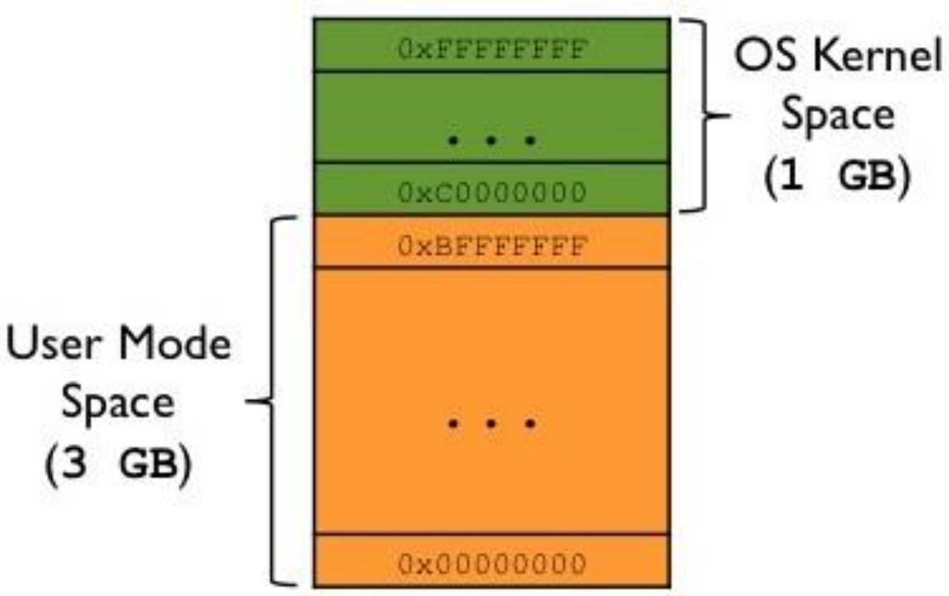
EECS 442: Computer Vision

- Very similar to Machine Learning but with a focus on image processing.
- Applications include autonomous driving and recognizing distinct objects in images.
- Less workload than 281, 6 homeworks, 1 project, 2 exams.

EECS 482: Operating Systems

- Continue to use C++ for lower-level programs.
- Project topics:
  - Concurrent (thread-based) programming
  - Virtual memory pager (from EECS 370)
  - Network file server
- Involves more thinking about system design and how to approach software problems.
- Workload is significantly heavier than 281 or 370, which is also a prereq.
- 4 projects, 2 exams.

EECS 482: Operating Systems

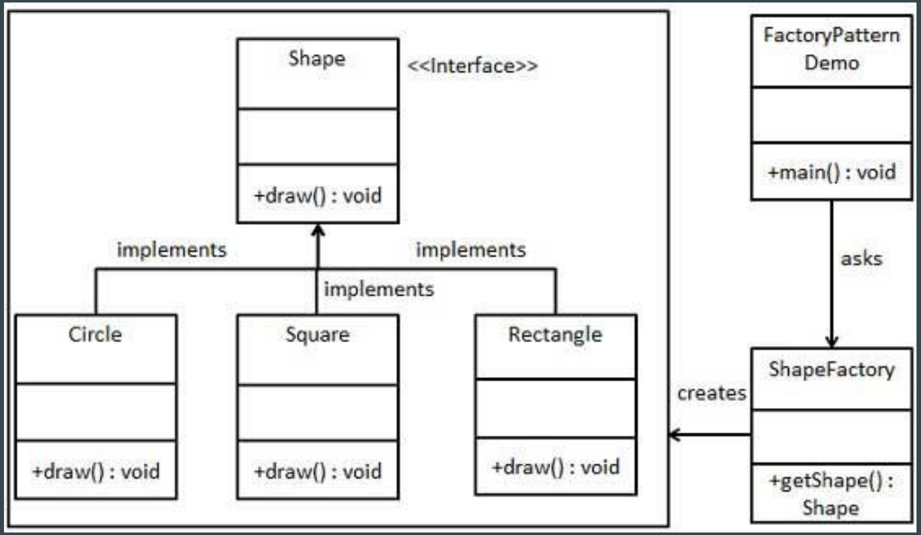


One of the many considerations of any good operating system: how to separate user process space from the kernel space.

EECS 381: Object-Oriented Programming

- Much more coding and less algorithmic focus than 281.
- C++ heavy, but focusing on general design concepts as well as C++ language features (std::shared_ptr, std::move, etc.).
- Around 12000 - 15000 lines of code – 7 projects total
  - Compared with 3200 - 4800 lines of code for 281.
- Professor hand grades about half of the projects
  - Graded for style, inefficiencies, duplicated code, overall code design, etc

EECS 381: Object-Oriented Programming

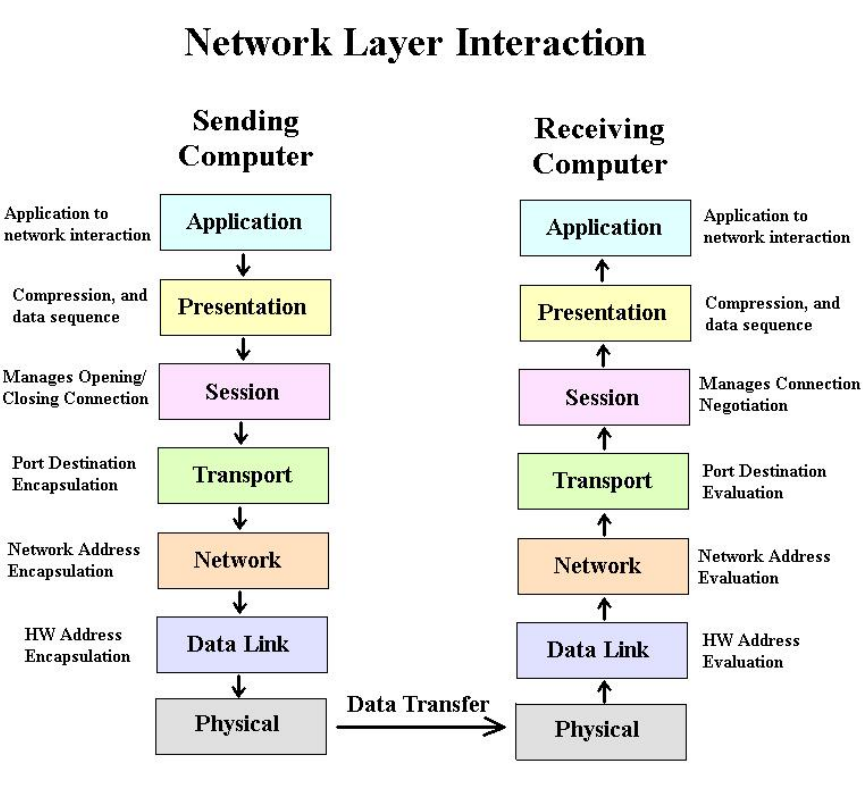


The factory design pattern: one of the many design paradigms discussed in EECS 381.

EECS 489: Networks

- Dig deep into how exactly the internet is implemented.
- Understand the different processing layers that data goes through when being sent through a network to facilitate secure communication.
- Discover what a packet actually is.
- Think about how to distribute a single source of information (website names and their respective IP addresses, for example) among many computers across the world.
- Workload: comparable to 280.

EECS 489: Networks

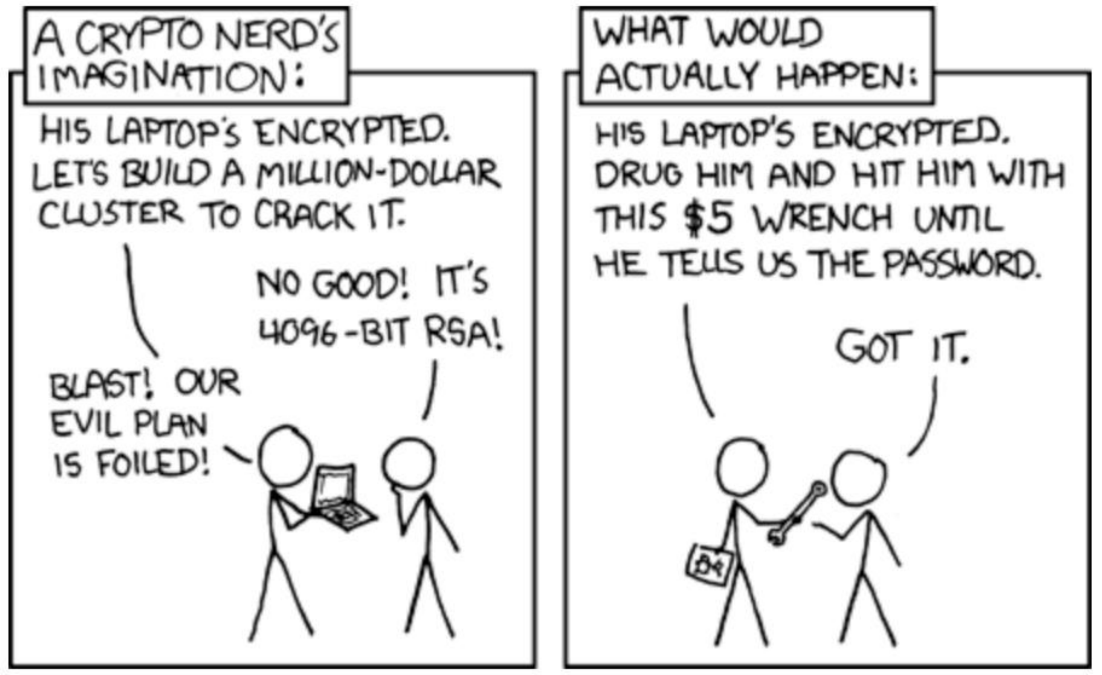

Network Layer Interaction
Sending Computer
Receiving Computer
Application to network interaction
Application
Compression, and data sequence
Presentation
Manages Opening/ Closing Connection
Session
Port Destination Encapsulation
Transport
Network Address Encapsulation
Network
HW Address Encapsulation
Data Link
Physical
Data Transfer
Application
Application to network interaction
Presentation
Compression, and data sequence
Session
Manages Connection Negotiation
Transport
Port Destination Evaluation
Network
Network Address Evaluation
Data Link
HW Address Evaluation
Physical

Networks = onions. Lots of layers.

EECS 475: Cryptography

- Starts with classical cryptography
  - Various ciphers such as Hill, Vigenere, Playfair
- Unit on abstract algebra (groups, rings, etc)
- More modern cryptography
  - DES, RSA, AES, Diffie-Hellman, Elliptic Curve Cryptography
- Weekly homework and one project.
- More math based than code heavy.
  - Should either be comfortable or prepared to learn both linear and topics from abstract algebra.
  - EECS 376 is required and if you enjoyed the crypto unit here, you might enjoy 475

EECS 475: Cryptography


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

https://xkcd.com/538/

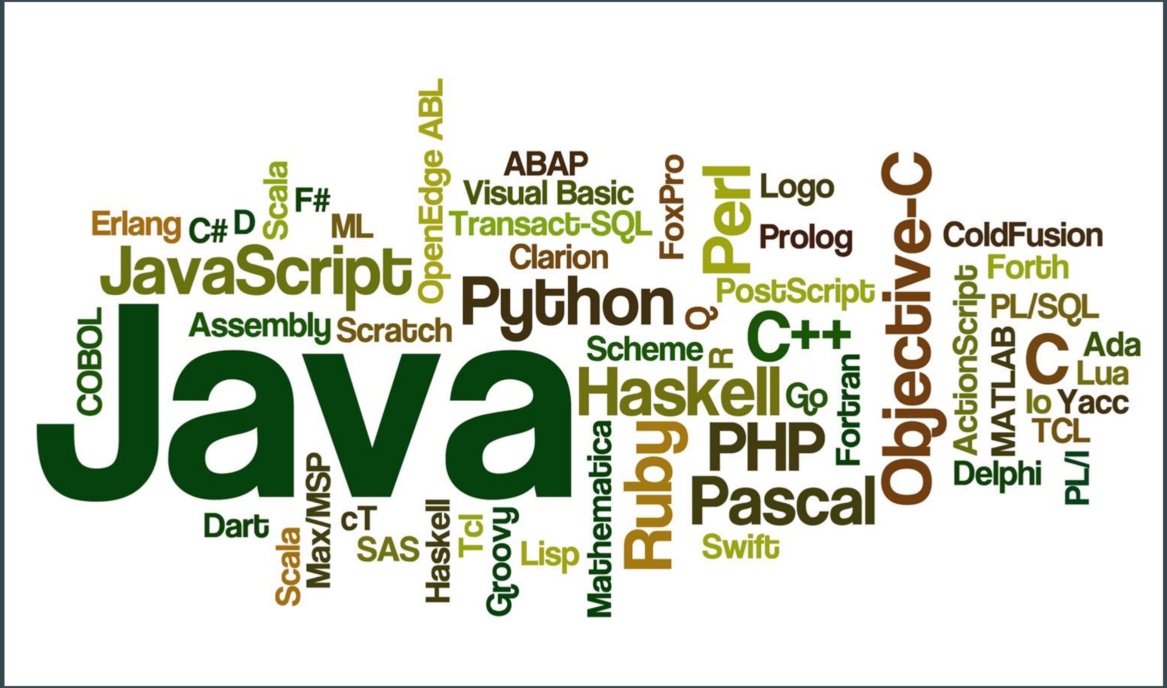
In 475, you will learn the internals of RSA encryption and how it works to protect the data that we send to each other.

EECS 477: Algorithms

- Focuses on the design and analysis (running time) of data structures and algorithms, including several advanced topics that will be more difficult to understand than those of 281.
- Assignments consist of several homeworks that will require writing mathematical proofs regarding correctness and time complexity.
- No required coding projects - a math based course.
  - Should be very comfortable with 281 material, as the course often builds off that.
  - EECS 376 is also a prereq, and the class will feel similar to that class in terms of the class structure and assignments, along with the theoretical side of 281. The workload is heavier than 376.

EECS 490: Programming Languages

- Not just about learning random programming languages, though you will learn a few, including some that are very different from C++.
- Covers theory of programming languages and their implementation, and how their features can best be used to their potential.
- Lighter workload than 281



EECS 398: Computing for Computer Scientists

- Spend more time with Darden
- Get a sweet looking (and more useful) command line.
- Learn how to use git!
- “Grow some bash-fu” - Darden
- Will be EECS 201 starting next Winter (2019).
- Learn a lot of skills that translate directly to workflow and working in industry or research!

And many others

- EECS 486 - Information Retrieval and Web Search - covers how useful information is taken from a body of information, such as text, and similarly how the web can be searched and organized into a useful result based on a query. Project-based (Python), lower workload than 281.
- EECS 481 - Software Engineering (new in Winter 2018) - focuses on how to structure, design, and analyze software, and teaches about relevant tools. Project-based, lower workload than 281.
- EECS 483 - Compiler Construction - How do we convert C++ code into something a machine can understand effectively, and how can we repeat this for other programming languages? Project-based (C++), generally lower workload than 281. 370 is also a prereq.
- EECS 492 - Artificial Intelligence, EECS 493 - User Interfaces, and some newer software upper levels just assigned course numbers now: e.g. EECS 491 - Distributed Systems. Many EECS 498 sections too, with classes like Data Mining and Natural Language Processing often being offered
- Hardware classes - some count as CS upper level classes too!: EECS 470 - Computer Architecture (how to implement a computer, from logic circuits to low-level software), EECS 373 - Embedded Systems, EECS 478 - Logic Circuit Synthesis and Optimization, EECS 427 - VLSI Design I