Programming

* Algorithms
  + Systematic way to solve problems
  + Flow charts
    - Oval
      * Terminal
        + “Start” and “end”
    - Rectangle -sequencing
      * Has step inside it, forming sequencing
    - Diamond -selection
      * Decision
      * 2 paths coming out of it, one for if condition inside is met, the other is for when it’s not met
    - Parallelogram
      * Data input or output
    - Connected with lines that has arrow pointing to next object
    - Iteration -repetition
      * Flow keeps on flowing back to a diamond for a while
* Register
  + Logs down variables and their values
* Display
  + Outputs program visually
  + Operated by graphics
* Compiler
  + Reads code
* Error checker
  + Ensures no error arises from execution of code
* Memory writer
  + Stores and edits into register
* Memory reader
  + Retrieves values from register
* Graphics
  + Displays and changes what’s on display
* Debugging
  + To check that everything’s working as it should
* Commenting
  + To note what a chunk of code does
* Pseudocode
  + Code written for human interpretation
* Abstraction -simplifies things
  + Procedural
    - When a chunk of code is to be repeated, define it as a function so it can be called simply by the function, without knowing how it works
  + Abstract data type
    - Data type not inbuilt to language
  + Simulation: generates new knowledge without performing actual thing
  + Higher level abstractions (most abstracted) make use of lower-level abstraction (least abstracted) on the assumption that it works, without concerning about how
* Snap
  + snap.berkeley.edu
  + Sprites
    - Objects on stage, manageable in corral
    - Costumes
      * Add by dropping it to costume tab
      * Modify with brush icon
  + Stage
    - display
    - Background
      * Drop pic to background tab
    - Size: 480 width by 360 height by default
  + Cleanup
    - In right click menu
    - Organizes blocks
  + Comments
    - Right click option to add them
  + Debugging
    - Visible stepping highlights code as it’s executed
    - Show variable shows values
  + Custom blocks
    - abstraction
    - Create block with make block
    - Basically compresses a chunk into just a block
    - Edit by right click
    - + sign can add inputs, so arguments can be inputted and passed to the algorithm within
    - Command makes procedural abstraction, as it involves steps. Reporter and predicates create abstract data type, as it reports values.
  + More blocks can be imported from files tab
  + Motion
    - Moves sprite
  + Looks
    - Changes sprite’s appearance
  + Sound
    - Makes sounds
  + Pen
    - Makes drawings
    - Pen down makes sprite leave a trail when moving
  + Controls
    - Creates conditionals, loops, etc for chunks of code
  + Sensing
    - Receives data
  + Operators
    - Does math and booleans
    - For random: if input is integer, outputs integer. If float, out is float
    - Boolean: And requires all to be true to be true, or requires at least one true to be true. Not negates truth value
  + Variables
    - Stores processable values
    - Global: accessible by all scripts
    - Local: accessible by certain scripts
  + Command
    - The bar with groove
    - Steps
  + Reporter
    - Oval
    - Reports a value
  + Predicate
    - Diamond
    - A decision, reports a Boolean
* Iteration: repeating
* Selection: code executed depends on case
* Sequence: series of steps
* Math: +-\*/ operators
* Logic: Booleans, AND (all needs to be true for compound to be true), OR (requires at least one to be true true for compound to be true), NOT (negates truth value), (exclusive or: both must be of different truth values for compound to be true)
* Counter: value that increases every iteration
* Recursive: function that calls on itself
* Problem decomposition: solve subproblems as stepping stones to main problem

Data

* Data types
  + Primitive data type
    - Inbuilt to language
    - In Snap, they are lists, numbers, Booleans, texts, sprites, commands, reporters and predicates
  + Abstract data types
    - Not inbuilt to language
    - In Snap, they are custom reporters and predicates
  + Variable
    - Stores data
    - Has to be initialized
    - Global: accessible to all scripts
    - Local: accessible to specific scripts
  + List
    - Stores multiple data points
    - May be stored into variable
  + Numbers
    - Integers: integers
    - Floats: decimals
    - (operation) Modulo: remainder after division
  + Booleans: true/false
* Big data
  + Very large, complex data set
  + Too many data points to be meaningful for as-it-is analysis
  + Too many data points for a human to process and manage
  + May contain private info such as names, addresses, etc
  + Computing power for processing big data is usually crowdsourced
  + Requires good visualization (sizes, shapes, colors, etc) and organization to provide meaningful insight
  + Can provide insight into trends, patterns, connections
* Data representation
  + Number systems
    - Place values work like this: rightmost is base0, second rightmost is base1, third rightmost is base2 and so on
    - Converting to decimal (base 10)
      * Align the digits to correct place, and multiply the digit by its place value
      * Add the products
    - Converting from decimal
      * Keep dividing the number by destination base until the quotient is less than 1
      * Record the integer part of the quotient, storing the first quotient on the right, second quotient left of the first, and so on
      * Take the destination base modulo of these quotients, and write directly underneath the corresponding quotient
    - Number of possible combinations: basedigits
      * Tip: an increase by 10bits is about 1000x increase in possible combinations
    - When adding or subtracting, use the base as the number used for carrying over when necessary
    - Binary number system (base 2)
      * 0 and 1 are the digits used
      * Bits: binary digit
      * A group of 3 bits can represent an octal digit
      * A group of 4 bits can represent a hexadecimal digit
    - Octal number system (base 8)
      * 0, 1, 2, 3, 4, 5, 6, 7 are the digits used
      * 3 octal digits represent 1 bit
    - Decimal number system: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 are the digits used
    - Hexadecimal number system (base 16)
      * 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (means 10), B (means 11), C (means 12), D (means 13), E (means 14), F (means 15) are the digits used
      * 4 hexadecimal digits represent 1 bit
  + Hexadecimal color codes: Format: #RRGGBB, where RR is the hex number representing the intensity of red, BB for blue, GG for green
  + ASCII: 7 (standard) or 8 (extended) bit numbers each mapped to a different character
  + Unicode: 16-bit number each mapped to a different character
* Compression
  + Lossy
    - Used when file size is a constraint
    - A way to format data in which it can no longer be restored to its original form, leading to details loss
  + Lossless
    - Used when details are a constraint
    - A way to format data in which it can still be restored to its original form

Internet

* *Inter*connected *net*work of computers and servers (nodes) around the world, governed by open protocols
* Internet Abstraction Hierarchy
  + Application layer protocols
    - Interpret and display data
    - HTML
      * Hypertext markup language
      * The code of the website
    - CSS: design
    - JavaScript: website functionality
    - HTTP
      * Hypertext transfer protocol
        + Protocol of formatting data (GET, POST requests)
      * Images are separate files, requiring separate HTTP requests, slowing down page loading
      * Not safe compared to HTTPS
    - DNS
      * Domain name system
      * Looks up IP address of domains for computer to connect to, based on the inputted URL
      * Hierarchy maintained by ICANN Internet Corporation for Assigned Names and Numbers
    - SMTP
      * Simple mail transfer protocol
      * Delivers emails
      * Message moves from sender’s email server’s outbox to receiver's email server’s inbox
  + Transport layer protocols
    - Breakdown data into packets, transfer and reconstruct
    - TCP
      * Transmission Control Protocol
      * Guarantees delivery of packets for fault tolerance
        + If all packets are successfully delivered, TCP will sign off and reconstruct data. Else, it won’t sign and will re-request missing/corrupt packets
    - UDP
      * User Datagram Protocol
      * No delivery guarantee
      * Used when low latency is important
  + Internet layer protocols
    - Manages pathway for packets to travel
    - IP
      * Internet protocol
      * Unique address of a node on a network
      * Hierarchy
        + Country.region.subnetwork.device
      * IPv4: 32 bits, 232 combos
        + Format: x.x.x.x, where x is a number that can be represented by 8 bits
        + Running out of unique addresses, so one address per subnetwork, not one per device, until transition to IPv6
      * IPv6: 128 bits, 2128 combos
        + Format: xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx, where x is a hex number
      * Local IP
        + Used to locate devices on a local network
        + Reserved ranges are 192.0.0.0-192.168.255.255, 10.0.0.0-10.255.255.255, 172.16.0.0-172.31.255.255
      * Public IP
        + Used to locate a local network
        + Associated with address, and other personal info
      * Allocation of addresses is maintained by ICANN
  + Network interface hardware/link layer
    - Physical connection between device and local network
    - Cables
      * Ethernet
    - Radios
      * WiFi
* Protocols
  + Standard of communication
  + HTML
  + HTTP
  + IP
  + TCP
  + UDP
  + SMTP
  + Internet protocols are open standards to ensure interoperability, not owned by any entity
  + Maintenance for protocols done by Internet Engineering Task Force and Internet SOCiety
* URL
  + Uniform resource locator
  + Web address
  + Hierarchy
    - Protocol://subdomain.root domain/path
* Browser software used to interpret HTML, CSS, JS, etc to make website usable
* Cookies: Stored in browser as text to identify an user
* Data is broken down and sent as packets
* Redundancy and fault tolerance
  + Multiple pathways for packets to travel through incase a node is down
  + More nodes, redundant nodes improve reliability
  + The ability to resend packets
* Abstraction
  + An IP can be represented by URL
  + Content on server can be represented by IP and path, without concerning what the content is
  + Abstract layers guarantee that they work without specifying how
  + Higher levels utilize lower levels without concerning how it functions
* Router: manages traffic and connects subnetwork to network
* End to end principle: routers only know it’s transfering data and doesn’t know what special treatment to give to different types of data
* Packet switching: breaking data into packets that can be individually delivered
  + Allows data to pass through allocated bandwidth (number of bits that can pass in a given moment)
  + Prevents everything from being lost in the event of a failure of a node
* Cybersecurity
  + Cryptography/Encryption
    - Scrambling a message in a way that can only be restored to original using key
    - Protocols: SSL, TLS, found in HTTPS
    - Symmetrical: both endpoints use the same key
    - Asymmetrical: Endpoints use different keys: a private that’s not shared and a public that’s shared
      * Public key shared so incoming content is encrypted with it, so that only private key of intended receiver can decrypt it
      * In TLS, public key encryption is used to share a symmetrical key to save time
    - Bruteforce: to test all possible keys
      * More bits makes it harder by increasing possible combos
    - There should be no patterns for security truly secure
    - Cipher: algorithm/key for generating/solving code
      * Shift cipher: moving all letters forward a set amount of letters (Caesar cipher)
        + To encrypt: a = 0 … z = 26

Then y =

Where y is position of final letter

X is original letter’s position

K is constant to shift by

% Modulo is returns the remainder after division

Make + to - to decrypt

* + - * Polyalphabetic cipher: each letter moved by different amount of letters
      * Bit Masking: convert data into binary, and perform bitwise (and, or, not, xor) on original using a mask of random binary. XOR most secure
      * Multiply 2 large prime numbers and it’s very hard to factor for original numbers
  + Attacks
    - Virus
      * Malware that causes computer to function in ways unexpected by user, such as nuances, damage, data mining, etc.
      * Spread by software vulnerabilities, peer to peer interactions and poor digital hygiene
    - Botnet
      * A large network of devices “cyberarmy” controlled by a person to remotely perform malicious activities such as DDoS attacks using devices on botnet
    - DDoS distributed denial of service
      * When a botnet spams a server with requests, overloading the server to the point it becomes inaccessible
    - Phishing
      * Tricking the user, such as getting them to input credentials into a decoy such as fake website
    - To avoid attacks, keep software up to date and don’t open questionable files and links
    - Information security CIA triad
      * Confidentiality
        + Prevent data leaks by keeping secrets inaccessible to those who aren’t authorized
      * Integrity
        + Prevent ransomware from overwriting files by preventing unauthorized writes
      * Availability
        + Prevent DDoS and ensure data access is timely and reliable
    - AAA service
      * User provide identity
      * Identity is authenticated with passphrase to prove identity
      * ID claim and proof are verified before authorization
      * Users have accountability for their actions
      * Actions done by user are non repudiable
* Repercussions
  + Internet creates issues such as social isolation if overused
  + Internet grants people the power to find information, but the ability to be censored and monitored too
  + Many jobs can become oi, replacing humans
  + On demand models of jobs would have less benefits than traditional employees
* Digital privacy
  + data published publicly intentionally or intentionally creates a person’s digital footprint, that can be used for or against a person
* Cyberbullying
  + Digital bullying
  + Bullies feel physically and identically protected by the Internet, making them more aggressive
  + Cyberbullying messages may be reposted and spread quickly
    - Not everyone who reposted will take it down
  + Ignore bully, save and turn over evidence
* Censorship
  + Blocking content
  + Censorship can come from government, or locally, such as employer
  + Specific domains may be blacklisted
  + Some connections may be blocked if specific key work is found
  + Politics that don’t align to country’s belief may be blocked
  + Social sites that can have offensive content may be blocked
  + Tools that can circumvent censorship may be blocked
  + Censorship violates human right to info
  + Censorship in a country may affect flow of info in, out or within another
* Search engines
  + Searches internet for content and sort them
  + A crawler analyzes pages by following hyperlinks. Data is stored onto index
  + Index are shown in results, ranked by publisher’s authority, freshness, number of times referenced, relevance to keyword, user, etc
  + Search results affected by many personal “signals” such as device, location, browsing history
  + Search engines typically store info about user, such as queries, timestamp, links clicked, queries inputted, location, device and browser fingerprints.
  + Search engines may send this info to links opened via a referral
  + Some search terms may be censored due to government demand
  + Search trends can be used to predict stuff such as when something becomes captures public attention, but not necessary interest

Algorithms

* Algorithm: systematic way to solve a problem
* Asymptotic notation
  + Used to indicate how long algorithm takes and how much resources used
  + Runtime: number of steps taken
  + Big theta
    - Represents realistic running time
  + Big O Not what you thought
    - Represents worst-case scenario running time
  + Big omega
    - Represents best case scenario running time
  + Lowest runtime is most efficient
  + Each of these notations is represented by an equation (constant, linear, quadratic, logarithmic, exponential, cubic, etc) in terms of n (number of elements in the list). Equation returns runtime
  + Runtime is considered reasonable when the runtime is less than a polynomial time (no more than nk where n is number of elements and k is constant)
  + Runtime is considered unreasonable when the runtime is a superpolynomial (more than nk, such as n!, kn)
  + Heuristics: when an approximate solution is acceptable and an exact solution would take an unreasonable amount of time to compute
  + A problem is solvable when there’s an algorithm that can solve it in a reasonable amount of time
  + A problem is unsolvable when there isn’t an algorithm that can solve it in a reasonable amount of time
  + A problem is decidable when there’s an algorithm that can solve it in all scenarios
  + A problem is undecidable when there isn’t an algorithm that can solve it in all scenarios
* Linear search
  + In an unsorted list guess one by one in order to find something
  + Big-O: n, big omega: 1
* Binary search
  + In a list sorted in order, guess middle, and determine if it’s too high or too low. Eliminate the impossible half and repeat with remaining half, so on
  + Big-O: log2n, big omega: 1
* Hash search
  + Put item in question into a rule (hashing) to get hash
  + If data is put into locations by hash, search in the hash’s section
  + Big-O: n, big omega: 1
* Bubble sort
  + Compare an element to the element in previous index. If previous element is larger, swap index.
  + Big O: n2, big omega: n
* Insertion sort
  + In a set of unordered numbers, from left to right, if number is less than numbers on left, move it left to where it belongs.
  + Big O: n2, big omega: n
* Breadth-first search
  + To find shortest path in graph
  + Starting at 0 from end (source) vertex, in each next vertex, add 1 to previous number to get its number. When there’s a fork, put next number at all possible next vertex, then continue the counting at each fork.
  + From start vertex, the algorithm will only go to the vertex that has a number less than its current number, to get shortest route.
  + Source vertex is indicated as null, to indicate that it has no predecessors
  + Big O: vertices + edges
* Merge sort
  + Big everything: nlog2n
  + Divide problems into subproblems, then each subproblem into sub-subproblems, solve sub-subproblems to solve subproblems, so that problem can be solved
* Quicksort
  + Big O: n2, other notations: nlog2n
  + In an array of unsorted numbers, take last number (pivot) and place it anywhere. If any numbers left of pivot is greater than pivot, send to right. This creates subarrays. Repeat process for subarrays until array is ordered.
* Selection sort
  + In a set of numbers not ordered, take smallest number and swap position with the first number in array, then second smallest with second position, and so on
  + BigO: n!, other notations: n2

Models and Simulations

* Model: computerized representation of object or system
* Simulation: put model through simulated conditions that the real object may encounter in real life
* Allows for experiments that may be expensive, time-consuming, dangerous and/or unethical to perform.
* Depends on abstraction to simulate factors.
* Simulations may not be completely accurate, since it may be missing factors.
* Using an iterative design process can help improve accuracy based on experience.
* Repeat simulation and aggregate data for more accurate data

Computers

* Abstraction
  + From highest to lowest:
  + Software domain
    - Apps
    - Languages
    - Libraries
    - OS
  + Digital domain: comprehends binaries
    - Architecture (ex: amd64, arm64, i386): instructions the software sends to components, that the components would understand
    - Components (ex: CPU, GPU, RAM, ROM)
    - Integrated circuits: black rectangles on circuit board
    - Logic gates: implements Boolean functions
  + Analog domain: comprehends analog signals (smooth continuous waves), which has in-between values, unlike binary
    - Transistors
* Logic circuits
  + Made of logic gates processing and passing values
    - AND gate: both inputs need to be true to output a true
    - OR gate: at least one inputted value need to be true for output to be true
    - NOT gate: negates inputted value
    - XOR gate: requires a true and a false input to output true
* Bitstreams
  + A sequence of bits
  + Word: A sequence of a specific number of bits
    - Contains special bits that tell which data type it is
  + Width: The number of bits a processor can handle in a given moment
    - Half of the width is used for negative numbers, one for zero, and the rest for positive numbers
  + Overflow error: when the word length is longer than the width
    - Bignum: an integer represented with multiple words to avoid an overflow error
    - Floating point: used to store non-integers or very large numbers through rounding to the nearest power of 2 and using scientific notation to approximate it to avoid an overflow error
      * The binary representation of non-integers after the decimal point:
      * Format: the sign of mantissa + location of decimal point + mantissa (coefficient of scientific notation, rounded to the nearest power of 2) + base (10) + sign of exponent + exponent
    - Exact rations: represent non-integers with 2 bignums: one for the numerator, one for the denominator
    - Binary coded decimal: represent each decimal digit using 4 bits
    - Decimal floating point: BCD, but with 4 extra bits to represent which power of 10 to multiply the BCD part by
  + NaN: not a number, illegal computation
* Moore’s Law: the number of transistors in processors will about double about every 2 years

Other stuff

* Copyright: copying and redistributing works by other people is illegal
  + Fair use: copying is allowed for educational, parody and reviewing purposes. It’s allowed as long as it’s not designed as a replacement of the original, has different purpose and target audience, and doesn’t affect the original
* AI: programming to allow computers to solve problem that requires human logistics
  + Needs to be trained. Supervised is when inputs are explained to the computer, and unsupervised is when computers have to attempt to interpret the input without solution given. AI learns through machine learning, (ML is part of AI)
  + May have biases
  + Some applications:
    - Machine vision: allows computer to see
      * Applications: computational photography, autonomous vehicles, object search, bar/qr code scanning, facial recognition
    - Natural language processing: allows computers and humans to communicate through words
      * Applications: Voice dictation, digital assistants, text to speech
* War drives innovation, as rapid innovation is an advantage
  + Dual use technology: technology used in warfare that can be repurposed to benefit civilians
* Computing enables easier, long distance communications, interactions and collaborations with small and large audiences
* Computing makes finding information much easier, as well as easier to unfairly copy information
* Computing has applications outside the computer science field
* Computing can socially, economically and culturally affect everyone

More types of algorithms, not in AP CSP curriculum:

* + Recursion
    - Works similar to the reversal of factorial
      * (Ex: 3! = 3 \* 2!, 2! = 2 \* 1!, 1! = 1 \* 0!, 0! = 1. Substitute 3! = 3 \* 2 \* 1 \* 1 = 6)
    - Also has logic of xn/2 means xn = y \* y so xn/2 = y
    - Can also determine if word is palindrome by comparing first and last letter, second and second to last, and so on.
  + Towers on Hanoi
    - Takes 2n-1 steps, where n is total number of items
    - Where there’s 3 spaces to place stacks, larger item below smaller item, move stack to another space with the ability to only move item on top of stacks
  + Graph
    - Shows network
    - Items are vertex
      * Degree of vertex is amount of edges connecting to it
      * Vertex are numbered, starting from 0 in adjacent order, stored as V
    - Connections are edges
      * Number on edge is weight, usually representing some distance
      * Edges are represented [vertex 1, vertex 2, and more if present], stored as E
    - Directed means connections are biconditional
      * Conditional is undirected
        + Indicated with one way arrow
    - Items are adjacent if connected by same edge
    - A path is 3+ edges
      * A path that ends where it starts is a cycle
        + No intermediate vertex are repeated
      * Shortest path has lowest sum of weight
    - Connected
      * Each vertex has a path to each vertex
      * Else is free tree