[**Wolfram Alpha**](https://www.wolframalpha.com/examples/science-and-technology/computational-sciences/)

[The Annotated Touring](https://www.amazon.com/Annotated-Turing-Through-Historic-Computability/dp/0470229055)

<https://www.algoexpert.io/product>

## **Programming:**

Text to speech service:

<https://ttsmp3.com/register?click=direct>

**WoW bot project:**

<https://www.ownedcore.com/>

<http://gameprogrammingpatterns.com/>

<http://gameprogrammingpatterns.com/game-loop.html>

<https://drewkestell.us/Article/6/Chapter/2>

<https://en.wikibooks.org/wiki/X86_Assembly>

<https://en.wikibooks.org/wiki/X86_Disassembly>

Software performance testing

Memory protection: Abstraction and arbitration

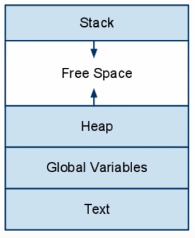
OS must ensure that a process cannot access memory that does not belong to it

Segmentation

valid/invalid pages

https://www.devdungeon.com/

**Memory Stack:**



Git research:  
Ssh.exe comes with git

GIT\_SSH and SVN\_SSH

tortoise(plink)

Account script:

Check acct balances and put them into a spreadsheet. Notifies me (by email) when the amount has changed. Checks daily at the same time.

A number keypad that lights up random numbers and you have to press them in the correct sequence (GUI)

Organize your quotes doc and create an app that randomly selects a different quote each morning

Machine learning

High temp and low temp searches

Google no longer supports “wildcard” entries to allow Flash

C++:<https://www.onlinegdb.com/online_c++_compiler>

Python:<https://repl.it/languages/python3>

Web Dev:<https://codepen.io/pen/tour/welcome/start>

Create a script that updates home server with USB changes when you plug it in

<https://atom.singularity2050.com/>

Relational model as opposed to object-relational

https://github.com/grammarly

research python to javascript translators for web apps

validator.w3.org – developer code compliance (low number of errors is ideal)

Research tracking pixels

add random() to subnetting program

Build subnetting gui

Design a turnkey solution for niche businesses

create an open source version of proprietary software

**Programming:**

Job control language (JCL) scripting languages used on IBM mainframe operating systems to instruct the system on how to run a batch job or start a subsystem.

Learn .NET framework

**programming language categories:**

Connection: internet or a network access between systems

Storage: a way to stop and retrieve information (backend, databases)

Interface: UI or some raw data protocol for input/output

**Take on an MVC project**

**Code to read:**

* UNIX V6 was the Unix originally written by Ken Thompson and Dennis Ritchie. Today it is pretty simple when compared with our current OSs. Anyway, it is the base for a bunch of related software nowadays. For sure that this code was read by the people which today is ruling the OS development world. I mean, people what is producing interesting code in this area…
* TeX is the equivalent of guttenberg’s press in computing, however you need to use it to understand why.
* Linux is one of the well succeeded open source projects and have shown the way of doing it (development, maintenance, etc).
* FreeBSD another example of an awesome open source project, btw, older than Linux. There is a bunch of interesting ideas too.
* Minix have been a good doorway for anyone interested in probing what is happening under the hood.
* Git is different and really makes scm easy and independent. It is written for/to people which love to code and love to work fast in a non-bureaucratic way. You are the owner of your copy and can work by yourself. Awesome!
* Plan9 was written by a dream team of coders. I really do not know what would be CompSci without Bell Labs. Ha-ha!

GNU Emacs. FORTRAN, the oldest programming language still in use, is pretty amazing especially for its time. Algol 60 is worth a mention if only because I still hold up the [Report on the algorithmic language ALGOL 60](http://dl.acm.org/citation.cfm?id=367262) as one of the finest pieces of technical writing ever.

Elastic Audio.

seam carving in digital image processors

Clan is the compiler for C?

**META II** is a [domain-specific](https://en.wikipedia.org/wiki/Domain-specific_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) for writing [compilers](https://en.wikipedia.org/wiki/Compiler).

**IDEs / code editors**

Electron Atom: Build cross-platform desktop apps with JavaScript, HTML, and CSS

<https://www.codeproject.com/Articles/866143/Learn-MVC-Project-in-> [days-Day](https://www.codeproject.com/Articles/866143/Learn-MVC-Project-in-days-Day)

in order to build a mobile app, you start by outlining the features the app will have and prioritize them

a programming projects agency that hires freelance programmers

website wireframe

website sitemap

**Learn to understand all of Habitica’s code and find all the ways that it can be exploited**

Join a programmer’s guild / party (find challenges)

Verilog and vhdl

[Project Based Learning Github projects](https://github.com/tuvtran/project-based-learning" \l "python)

Natural language processing language programming

A Time App. (it’s not called clock, it’s called TIME)

multiple countdowns

Alarms

Interval timers with saved presets

Multiple stopwatches at a time, with labels

* Stopping one starts the other and vice versa

**Research if you can put your own app on your phone without putting it up on the app store**

**MOVE YOUR FUCKING PERSONAL FILES OFF GOOGLE DRIVE AND ONTO A PERSONAL SERVER ON YOUR HOME NETWORK**

**SECURE YOUR MOTHERFUCKING HOME NETWORK**

**If you have to explain the project to someone before they can use it, you have a horrendous UI/UX. Projects should be so intuitive that a person can understand what they’re looking at just by interacting with it**

Building a web scraper that automates tasks on websites you frequently visit

Build an API with an SQL database

Build a simple snake game

Build a web server to replace google drive and use as a backup

Create a CHIP-8 emulator

**In order to be a good programmer, you need an insatiable curiosity**

**Characteristics of coding projects that will land you a programming job**

Build projects in the MVC design pattern. It is a standard in the industry

Project needs to have a database with CRUD: create, read, update, delete

Needs a logon page, authentication and authorization: auth0

Use a bootstrap template (needs a strong UI)

Make sure is solves a business problem

DLL injection, assembly injection, interoperability, function detouring, and interacting with the Windows API

How do pointers and memory management work (C++)

Grep and awk

Research how to crack software

all AI papers are trash: ADD MORE LAYERS!

[Bypass Paywalls](https://github.com/iamadamdev/bypass-paywalls-chrome/blob/master/README.md)

Seguin Linux

**Web scraping project needs randomized User Agents that bypass security**  
<https://www.scrapehero.com/how-to-fake-and-rotate-user-agents-using-python-3/>

[Multiple API Documentation (DevDocs)](https://devdocs.io/)

[BeautifulSoup documentation](https://www.crummy.com/software/BeautifulSoup/bs4/doc/)

[Introduction to XML](https://www.sitepoint.com/really-good-introduction-xml/)

<https://www.amazon.com/Design-Patterns-Object-Oriented-Addison-Wesley-Professional-ebook/dp/B000SEIBB8>

SRS: software, requirement, specification

.NET, C#, asp.net MVC

Bug tracker program demo: <https://www.youtube.com/watch?v=vG824vBdYY8>

Quines

What does one need to know in order to call himself a full stack web programmer?

The biggest problem with tutorials is that they become outdated by the time they are posted and they don’t actually teach you how to be a programmer, they teach you how to follow a set of sets. That’s not how programmers think. Programmers think abstractly and design software around a particular set of goals. You have to stop teaching people how to code and start teaching people how to think and code like a programmer.

[Violent Python](https://www.amazon.com/Violent-Python-Cookbook-Penetration-Engineers/dp/1597499579)

[Blackhat Python](https://www.amazon.com/Black-Hat-Python-Programming-Pentesters/dp/1593275900)

[Grayhat Python](https://www.amazon.com/Gray-Hat-Python-Programming-Engineers/dp/1593271921)

One free week of code academy

Build a ping sweeper in python

Kali linux: chmod +x [python\_script\_name].py

Ackerman's function

Write a buffer overflow script

Python port scanner

[2020 Stack Exchange Developer Survey](https://insights.stackoverflow.com/survey/2020)

Learn Rust programming language

Learn the asp.net core web framework

Build an interval timer app (android)

[Python cheat sheet](https://darknetdiaries.com/python/)

## **Programming Goals:**

take a data structures course

become CTO for a startup

Write a floating point package

go to a hackathon

Explore python and C# documentation by writing small programs for methods

Take a data science course

Get through <https://codingbat.com/prob/p173401> in both java and python

## **Skills to learn:**

PHP

mySQL

get better at reading other people’s code

linux command line

windows CMD

javascript

github

Golang

scripting

machine learning

## **Coding Projects:**

* [Explore the Binary Game](https://studio.code.org/projects/applab/iukLbcDnzqgoxuu810unLw/view)
* Build a subnetting table: takes an IP address with CIDR and solves for all 7 attributes
* Build Python Blackjack game
* Create vlsm CIDR subnet calculator: [Vlsm-calc.net](http://vlsm-calc.net/)
* Build a website with Python
* Build a random quote machine (or tao te ching passages) in Python
* sudoku solver
* pathfinding algorithm
* calculator
* write a program that calculates how many seconds, minutes, hours, days, weeks, months, years, decades in the past a date was

**Evernote-esque system for Linux**

* Allows you to encrypt files
* Save/work locally
* Google docs functionality
* PDF saver/highlighter/note taker
* Add tags

## 

## **Numbers:**

**Find PI to the Nth Digit** - Enter a number and have the program generate PI up to that many decimal places. Keep a limit to how far the program will go.

**Find e to the Nth Digit** - Just like the previous problem, but with e instead of PI. Enter a number and have the program generate e up to that many decimal places. Keep a limit to how far the program will go.

**Fibonacci Sequence** - Enter a number and have the program generate the Fibonacci sequence to that number or to the Nth number.

**Prime Factorization** - Have the user enter a number and find all Prime Factors (if there are any) and display them.

**Next Prime Number** - Have the program find prime numbers until the user chooses to stop asking for the next one.

**Find Cost of Tile to Cover W x H Floor** - Calculate the total cost of tile it would take to cover a floor plan of width and height, using a cost entered by the user.

**Mortgage Calculator** - Calculate the monthly payments of a fixed term mortgage over given Nth terms at a given interest rate. Also figure out how long it will take the user to pay back the loan. For added complexity, add an option for users to select the compounding interval (Monthly, Weekly, Daily, Continually).

**Change Return Program** - The user enters a cost and then the amount of money given. The program will figure out the change and the number of quarters, dimes, nickels, pennies needed for the change.

**Binary to Decimal and Back Converter** - Develop a converter to convert a decimal number to binary or a binary number to its decimal equivalent.

**Calculator** - A simple calculator to do basic operators. Make it a scientific calculator for added complexity.

**Unit Converter (temp, currency, volume, mass and more)** - Converts various units between one another. The user enters the type of unit being entered, the type of unit they want to convert to and then the value. The program will then make the conversion.

**Alarm Clock** - A simple clock where it plays a sound after X number of minutes/seconds or at a particular time.

**Distance Between Two Cities** - Calculates the distance between two cities and allows the user to specify a unit of distance. This program may require finding coordinates for the cities like latitude and longitude.

**Credit Card Validator** - Takes in a credit card number from a common credit card vendor (Visa, MasterCard, American Express, Discover) and validates it to make sure that it is a valid number (look into how credit cards use a checksum).

**Tax Calculator** - Asks the user to enter a cost and either a country or state tax. It then returns the tax plus the total cost with tax.

**Factorial Finder** - The Factorial of a positive integer, n, is defined as the product of the sequence n, n-1, n-2, ...1 and the factorial of zero, 0, is defined as being 1. Solve this using both loops and recursion.

**Complex Number Algebra** - Show addition, multiplication, negation, and inversion of complex numbers in separate functions. (Subtraction and division operations can be made with pairs of these operations.) Print the results for each operation tested.

**Happy Numbers** - A happy number is defined by the following process. Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers. Display an example of your output here. Find the first 8 happy numbers.

**Number Names** - Show how to spell out a number in English. You can use a pre existing implementation or roll your own, but you should support inputs up to at least one million (or the maximum value of your language's default bounded integer type, if that's less). *Optional: Support for inputs other than positive integers (like zero, negative integers, and floating-point numbers).*

**Coin Flip Simulation** - Write some code that simulates flipping a single coin however many times the user decides. The code should record the outcomes and count the number of tails and heads.

**Limit Calculator** - Ask the user to enter f(x) and the limit value, then return the value of the limit statement *Optional: Make the calculator capable of supporting infinite limits.*

**Fast Exponentiation** - Ask the user to enter 2 integers a and b and output a^b (i.e. pow(a,b)) in O(lg n) time complexity

**Time Adder** - adds up multiple intervals of time in one unit and outputs them in another (eg how many minutes in 460 seconds or how many minutes is 1:45 + 6:08 + 3:02 + 0:45 etc)

## 

## **Classic Algorithms**

**Collatz Conjecture** - Start with a number *n > 1*. Find the number of steps it takes to reach one using the following process: If *n* is even, divide it by 2. If *n* is odd, multiply it by 3 and add 1.

**Sorting** - Implement two types of sorting algorithms: Merge sort and bubble sort.

**Closest pair problem** - The closest pair of points problem or closest pair problem is a problem of computational geometry: given *n* points in metric space, find a pair of points with the smallest distance between them.

**Sieve of Eratosthenes** - The sieve of Eratosthenes is one of the most efficient ways to find all of the smaller primes (below 10 million or so).

## 

## **Graph:**

**Graph from links** - Create a program that will create a graph or network from a series of links.

**Eulerian Path** - Create a program which will take as an input a graph and output either a Eulerian path or a Eulerian cycle, or state that it is not possible. A Eulerian Path starts at one node and traverses every edge of a graph through every node and finishes at another node. A Eulerian cycle is a eulerian Path that starts and finishes at the same node.

**Connected Graph** - Create a program which takes a graph as an input and outputs whether every node is connected or not.

**Dijkstra’s Algorithm** - Create a program that finds the shortest path through a graph using its edges.

**Minimum Spanning Tree** - Create a program which takes a connected, undirected graph with weights and outputs the minimum spanning tree of the graph i.e., a subgraph that is a tree, contains all the vertices, and the sum of its weights is the least possible.

## 

## **Data Structures:**

**Inverted index** - An [Inverted Index](http://en.wikipedia.org/wiki/Inverted_index) is a data structure used to create full text search. Given a set of text files, implement a program to create an inverted index. Also create a user interface to do a search using that inverted index which returns a list of files that contain the query term / terms. The search index can be in memory.

## 

## **Text:**

**Fizz Buzz** - Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”.

**Reverse a String** - Enter a string and the program will reverse it and print it out.

**Pig Latin** - Pig Latin is a game of alterations played on the English language game. To create the Pig Latin form of an English word the initial consonant sound is transposed to the end of the word and an ay is affixed (Ex.: "banana" would yield anana-bay). Read Wikipedia for more information on rules.

**Count Vowels** - Enter a string and the program counts the number of vowels in the text. For added complexity have it report a sum of each vowel found.

**Check if Palindrome** - Checks if the string entered by the user is a palindrome. That is that it reads the same forwards as backwards like “racecar”

**Count Words in a String** - Counts the number of individual words in a string. For added complexity read these strings in from a text file and generate a summary.

**Text Editor** - Notepad style application that can open, edit, and save text documents. *Optional: Add syntax highlighting and other features.*

**RSS Feed Creator** - Given a link to RSS/Atom Feed, get all posts and display them.

**Quote Tracker (market symbols etc)** - A program which can go out and check the current value of stocks for a list of symbols entered by the user. The user can set how often the stocks are checked. For CLI, show whether the stock has moved up or down. *Optional: If GUI, the program can show green up and red down arrows to show which direction the stock value has moved.*

**Guestbook / Journal** - A simple application that allows people to add comments or write journal entries. It can allow comments or not and timestamps for all entries. Could also be made into a shout box. *Optional: Deploy it on Google App Engine or Heroku or any other PaaS (if possible, of course).*

**Vigenere / Vernam / Caesar Ciphers** - Functions for encrypting and decrypting data messages. Then send them to a friend.

**Regex Query Tool** - A tool that allows the user to enter a text string and then in a separate control enter a regex pattern. It will run the regular expression against the source text and return any matches or flag errors in the regular expression.

## 

## **Networking:**

**FTP Program** - A file transfer program which can transfer files back and forth from a remote web server.

**Bandwidth Monitor** - A small utility program that tracks how much data you have uploaded and downloaded from the net during the course of your current online session. See if you can find out what periods of the day you use more and less and generate a report or graph that shows it.

**Port Scanner** - Enter an IP address and a port range where the program will then attempt to find open ports on the given computer by connecting to each of them. On any successful connections mark the port as open.

**Mail Checker (POP3 / IMAP)** - The user enters various account information including web server and IP, protocol type (POP3 or IMAP) and the application will check for email at a given interval.

**Country from IP Lookup** - Enter an IP address and find the country that IP is registered in. *Optional: Find the Ip automatically.*

**Whois Search Tool** - Enter an IP or host address and have it look it up through whois and return the results to you.

**Site Checker with Time Scheduling** - An application that attempts to connect to a website or server every so many minutes or a given time and check if it is up. If it is down, it will notify you by email or by posting a notice on screen.

## 

## **Classes:**

**Product Inventory Project** - Create an application which manages an inventory of products. Create a product class which has a price, id, and quantity on hand. Then create an *inventory* class which keeps track of various products and can sum up the inventory value.

**Airline / Hotel Reservation System** - Create a reservation system which books airline seats or hotel rooms. It charges various rates for particular sections of the plane or hotel. Example, first class is going to cost more than coach. Hotel rooms have penthouse suites which cost more. Keep track of when rooms will be available and can be scheduled.

**Company Manager** - Create a hierarchy of classes - abstract class Employee and subclasses HourlyEmployee, SalariedEmployee, Manager and Executive. Everyone's pay is calculated differently, research a bit about it. After you've established an employee hierarchy, create a Company class that allows you to manage the employees. You should be able to hire, fire and raise employees.

**Bank Account Manager** - Create a class called Account which will be an abstract class for three other classes called CheckingAccount, SavingsAccount and BusinessAccount. Manage credits and debits from these accounts through an ATM style program.

**Patient / Doctor Scheduler** - Create a patient class and a doctor class. Have a doctor that can handle multiple patients and set up a scheduling program where a doctor can only handle 16 patients during an 8 hr work day.

**Recipe Creator and Manager** - Create a recipe class with ingredients and put them in a recipe manager program that organizes them into categories like deserts, main courses or by ingredients like chicken, beef, soups, pies etc.

**Image Gallery** - Create an image abstract class and then a class that inherits from it for each image type. Put them in a program which displays them in a gallery style format for viewing.

**Shape Area and Perimeter Classes** - Create an abstract class called Shape and then inherit from it other shapes like diamond, rectangle, circle, triangle etc. Then have each class override the area and perimeter functionality to handle each shape type.

**Flower Shop Ordering To Go** - Create a flower shop application which deals in flower objects and use those flower objects in a bouquet object which can then be sold. Keep track of the number of objects and when you may need to order more.

**Family Tree Creator** - Create a class called Person which will have a name, when they were born and when (and if) they died. Allow the user to create these Person classes and put them into a family tree structure. Print out the tree to the screen.

## 

## **Threading:**

**Create A Progress Bar for Downloads** - Create a progress bar for applications that can keep track of a download in progress. The progress bar will be on a separate thread and will communicate with the main thread using delegates.

**Bulk Thumbnail Creator** - Picture processing can take a bit of time for some transformations. Especially if the image is large. Create an image program which can take hundreds of images and convert them to a specified size in the background thread while you do other things. For added complexity, have one thread handling re-sizing, have another bulk renaming of thumbnails etc.

## 

## **Web:**

**Page Scraper** - Create an application which connects to a site and pulls out all links, or images, and saves them to a list. *Optional: Organize the indexed content and don’t allow duplicates. Have it put the results into an easily searchable index file.*

**Online White Board** - Create an application which allows you to draw pictures, write notes and use various colors to flesh out ideas for projects. *Optional: Add feature to invite friends to collaborate on a white board online.*

**Get Atomic Time from Internet Clock** - This program will get the true atomic time from an atomic time clock on the Internet. Use any one of the atomic clocks returned by a simple Google search.

**Fetch Current Weather** - Get the current weather for a given zip/postal code. *Optional: Try locating the user automatically.*

**Scheduled Auto Login and Action** - Make an application which logs into a given site on a schedule and invokes a certain action and then logs out. This can be useful for checking web mail, posting regular content, or getting info for other applications and saving it to your computer.

**E-Card Generator** - Make a site that allows people to generate their own little e-cards and send them to other people. Do not use Flash. Use a picture library and perhaps insightful mottos or quotes.

**Content Management System** - Create a content management system (CMS) like Joomla, Drupal, PHP Nuke etc. Start small. *Optional: Allow for the addition of modules/addons.*

**Web Board (Forum)** - Create a forum for you and your buddies to post, administer and share thoughts and ideas.

**CAPTCHA Maker** - Ever see those images with letters and numbers when you sign up for a service and then ask you to enter what you see? It keeps web bots from automatically signing up and spamming. Try creating one yourself for online forms.

## 

## **Files:**

**Quiz Maker** - Make an application which takes various questions from a file, picked randomly, and puts together a quiz for students. Each quiz can be different and then reads a key to grade the quizzes.

**Sort Excel/CSV File Utility** - Reads a file of records, sorts them, and then writes them back to the file. Allow the user to choose various sort styles and sorting based on a particular field.

**Create Zip File Maker** - The user enters various files from different directories and the program zips them up into a zip file. *Optional: Apply actual compression to the files. Start with Huffman Algorithm.*

**PDF Generator** - An application which can read in a text file, html file or some other file and generates a PDF file out of it. Great for a web based service where the user uploads the file and the program returns a PDF of the file. *Optional: Deploy on GAE or Heroku if possible.*

**MP3 Tagger** - Modify and add ID3v1 tags to MP3 files. See if you can also add in the album art into the MP3 file’s header as well as other ID3v2 tags.

**Code Snippet Manager** - Another utility program that allows coders to put in functions, classes or other tidbits to save for use later. Organized by the type of snippet or language the coder can quickly look up code. *Optional: For extra practice try adding syntax highlighting based on the language.*

## 

## **Databases:**

**SQL Query Analyzer** - A utility application which a user can enter a query and have it run against a local database and look for ways to make it more efficient.

**Remote SQL Tool** - A utility that can execute queries on remote servers from your local computer across the Internet. It should take in a remote host, username and password, run the query and return the results.

**Report Generator** - Create a utility that generates a report based on some tables in a database. Generates sales reports based on the order/order details tables or sums up the days current database activity.

**Event Scheduler and Calendar** - Make an application which allows the user to enter a date and time of an event, event notes and then schedule those events on a calendar. The user can then browse the calendar or search the calendar for specific events. *Optional: Allow the application to create recurring events that recur every day, week, month, year etc.*

**Budget Tracker** - Write an application that keeps track of a household’s budget. The user can add expenses, income, and recurring costs to find out how much they are saving or losing over a period of time. *Optional: Allow the user to specify a date range and see the net flow of money in and out of the house budget for that time period.*

**TV Show Tracker** - Got a favorite show you don’t want to miss? Don’t have a PVR or want to be able to find the show to then PVR it later? Make an application which can search various online TV Guide sites, locate the shows/times/channels and add them to a database application. The database/website then can send you email reminders that a show is about to start and which channel it will be on.

**Travel Planner System** - Make a system that allows users to put together their own little travel itinerary and keep track of the airline / hotel arrangements, points of interest, budget and schedule.

## 

## **Graphics and Multimedia:**

**Slide Show** - Make an application that shows various pictures in a slide show format. *Optional: Try adding various effects like fade in/out, star wipe and window blinds transitions.*

**Stream Video from Online** - Try to create your own online streaming video player.

**MP3 Player** - A simple program for playing your favorite music files. Add features you think are missing from your favorite music player.

**Watermarking Application** - Have some pictures you want copyright protected? Add your own logo or text lightly across the background so that no one can simply steal your graphics off your site. Make a program that will add this watermark to the picture. *Optional: Use threading to process multiple images simultaneously.*

**Turtle Graphics** - This is a common project where you create a floor of 20 x 20 squares. Using various commands you tell a turtle to draw a line on the floor. You have to move forward, left or right, lift or drop pen etc. Do a search online for "Turtle Graphics" for more information. *Optional: Allow the program to read in the list of commands from a file.*

**GIF Creator** A program that puts together multiple images (PNGs, JPGs, TIFFs) to make a smooth GIF that can be exported. *Optional: Make the program convert small video files to GIFs as well.*

## 

## **Security:**

**Caesar cipher** - Implement a Caesar cipher, both encoding and decoding. The key is an integer from 1 to 25. This cipher rotates the letters of the alphabet (A to Z). The encoding replaces each letter with the 1st to 25th next letter in the alphabet (wrapping Z to A). So key 2 encrypts "HI" to "JK", but key 20 encrypts "HI" to "BC". This simple "monoalphabetic substitution cipher" provides almost no security, because an attacker who has the encoded message can either use frequency analysis to guess the key, or just try all 25 keys

## **Programming Research:**

Paul Graham website

DOM: document object model

Learn UX and UI design

https://github.com/tgalal/yowsup

What is the difference between PHP and mySQL?

RDBMS like my sql: oracle, mysql, mssql and postgresql

good code vs bad code

What is transpilation?

Bootstrap templates

Exploring compilers

What is apache?

Flask / Jinja / Django

building a website with node.js and express.js

angular js: building a data driven app

php myadmin and regular php

difference between git and github

web project workflows

agile software development

netbeans for java

eclipse IDE

android studio

[**https://codegolf.meta.stackexchange.com/**](https://codegolf.meta.stackexchange.com/)

UTF-8 standard

Programming time zone

recursion

Javascript is an interpreted language? - types of programming languages  
different categories of programming

types of programming jobs

what is it like to be a freelance programmer

Website testing / scraping

browser developer consoles

Something awful forums

Conceptual programming with python

Emergent AI

artificial intelligence

raspberry pi projects

Python meetups and coding bootcamps / hackathons

[Heroku](https://devcenter.heroku.com/articles/git) and [Github Pages](https://pages.github.com/)

firebase python sdk

Directory Services

What happens when you compile a source file?

What would it mean to write your own compiler?

When you import or use external libraries, where do they come from?

What does it *actually* mean to import an external library?

How does your compiler, operating system, or virtual machine link multiple pieces of code together to form a coherent system?

What are code generators?

What are starter kits?

SKD packages for Linux, Android, and *Web Development*

Object Oriented Programming: *Liskov Substitution Principle*

Learn how to minify / concatenate CSS and JS files

Differences between scripting languages, functional languages, and high-performance/compiled languages

Topcoder.com

Gdb linux assembly

Mobile development, how does it differ from other software development?

What is a data scientist?

Code Kata PragProg

**superscripts**

Encapsulation

Abstraction

Inheritance

Polymorphism

Classes

Graphics Interactive **Programming** (**GRIP**) allows you to automate some operations in NX

**Javascript research:**

* Anonymous functions
* Newer ES6 syntax
* LABjs
* asynquence: <https://github.com/getify/asynquence>

**Ruby research:**

* [Ruby User Groups](https://www.ruby-lang.org/en/community/user-groups/)
* [Ruby From Python](https://www.ruby-lang.org/en/documentation/ruby-from-other-languages/)
* Ruby Getter / Setter methods

**We are better developers when we can more fully understand how our computer interprets out code**