List of Projects: *(project implementation available on github @pgopics upon request)*

ft. some projects i've done

Java:

- **Interest Table:**
  - Created an interest table calculator that calculated simple and compound interest up to 25 years. Developed and wrote code for the GUI and gained experience with JavaFX. *(inner/nested/anonymous classes, Lambda, JavaFX)*

- **Orders Processor:**
  - Wrote a program that processed any number of customer purchase orders. *(threads, multithreading, synchronization)*

- **Online Test:**
  - Wrote the data manager for an online test system to generate and grade exams with multiple question types. *(Object Oriented Design, Inheritance)*

- **Clear Cell Game:**
  - Implemented a game called Clear Cell Game, with the goal of maximizing the number of cleared cells in a colored board. *(abstraction, recursion, inheritance, encapsulation, JavaFX)*

- **Media Rental Manager:**
  - Implemented a system that allows people to rent movies and music albums (similar to Netflix) and receive them via mail. *(Object Oriented Design, Inheritance)*

Data Structures: (Java)

- **Linked List:**
  - Wrote various LinkedLists, Doubly Linked Lists, and SortedLinkedLists methods. Implemented a Java collection with a parameterized type. Wrote JUnit Tests for the project. *(LinkedLists)*

- **Binary Search Tree:**
  - Implemented a polymorphic binary search tree. *(Recursion, Polymorphism, Testing)*

- **Graphs:**
  - Implemented a undirected weighted Graph. Preformed Depth First Search, Breadth First Search, and Dijsktra's Algorithm to get the shortest distance & find the shortest path *(Graphs, Dijsktra's Algorithm, BFS, DFS)*

C:

- **Draw Figures:**
  - Wrote a program in C that draws a rectangle, triangle, and rhombus *(C basics, functions)*

- **Grades Calculator**
  - Wrote a program in C that reads assignments scores and computes grades and statistical information (mean and standard deviation) . Accounted for late penalty, dropped assignments, and assignment weight. *(Functions, Arrays)*

- **Document Manager**
  - Wrote a program in C to allow the user to edit a document. Allowed users to add paragraphs, add lines to paragraphs, and replace/highlight/remove text. *(Structs. String Manipulation)*

- **User - Interface:**
  - Created a text-based user interface to the Document Manager project. Parsed text from input data files for valid lines and commands. Additionally implemented code to allow users to load files and save documents. *(Text parsing, File I/O)*

- **Calendar:**
  - Created a Calendar application in C to allow scheduling of events at specific days. Wrote a makefile for this project as well as students test to test for edge cases/errors. *(Structs, Dynamic Memory Allocation, Function Pointers, Linked Lists)*

- **Shell Sr:**
  - Implemented a shell in C that read and executed valid commands. Shell supported boolean operations, conjunctions (“&&” operator), pipes, and file redirection, and subshells. Wrote a makefile for this project. *(parse trees, fork(), exec*, lexing/parsing)*

Assembly/MIPS:

- **C Translator:**
  - Wrote MIPS assembly code that corresponds to C Code. *(Assembly)*

Ruby:

- **Translator:**
  - Created a translator in Ruby that translates sentences in a variety of languages: English, French, German, Swedish, Spanish etc. Wrote a grammar class that assisted to keep in track of grammars. *(Nested Hashmaps, Regex, Functions).*

OCaml:

- **Database Design:**
  - Created a program that stored a person (name, age, hobbies) in a database. Queried the database given a condition.

- **Trees:**
  - Implemented higher order Ocaml functions on binary search trees without using recursion. Used fold function to write tree fold. Used map to write tree map. Wrote code for mirror tree, trim tree, depth tree, compose tree, in-order and preorder traversals. Lastly, given a list containing pre-order traversal of a tree and list containing the in-order traversal of a tree, created a tree that corresponds to the two traversals. *(Binary Search Trees, Higher Order Functions: Map/Fold)*

- **Regular Expression Engine (NFA/DFA):**
  - Implemented various algorithms to work with NFAs, DFAs, and regex: *accept* to see if a function is accepted by NFA, *subset construction* to convert NFA to DFA, and *regex_to_nfa* to convert regex to NFA. *(NFA, DFA, regex)*

- **Interpreter:**
  - Wrote mutop, an interpreter (lexer, parser, and evaluator) for Microcaml - dynamically typed version of Ocaml. *(Lexer, Parser, Evaluator, Abstract Syntax Trees)*

Rust:

- **Stark Suit Repair:**
  - Wrote a min heap data structure to write functions in the PriorityQueue trait for the vec Type. Implemented a program that returns the name and coordinates of the enemy Stark will fight. *(Traits, Vectors, Types, Min Heaps)*

Matlab:

- **Investment Portfolio:**
  - Matlab project that analyzed stock returns and optimized shares for maximum expected returns for Amazon, Biogen Inc., and Edwards Lifesciences Corp.

R:

- **electric gasoline vehicle research:**
    - Worked with various data frames from National Emissions Inventory. Calculated marginal damages in cents/mile for NOx and SO2 for both electric and gasoline vehicles. Modeled results by creating tables, graphs, and maps in R to detail NOx and SO2 emissions for different interconnections (NERC) in the US.