

CSPB 3112 Project Final Report

Rachel Mertz (rame4494)

8/12/24

Introduction

The immediate-term goal of my summer project was to build and maintain C++ proficiency to ensure a smooth transition from the basics covered in CSPB 1300 (Intro CS) to the more advanced object-oriented concepts to be covered in CSPB 2700 (Data Structures), which I plan to take this fall alongside CSPB 3104 (Algorithms) and CSPB 2820 (Linear Algebra). The medium and longer-term goals of this work are to take a full slate of advanced data science electives in the spring, and to complete a career pivot onto a computational biology track later in 2025. Although I anticipate that the majority of my work in my new role will use Python rather than C++, a strong foundational understanding of data structures in a compiled language will ultimately make me a better-rounded scientist with a deeper conceptual understanding of the underlying frameworks used in Python's machine learning libraries and a broader array of tools and approaches for troubleshooting.

Goals

My goals are presented as described in my original Project Proposal, with revisions from the Peer Review feedback Project Update Report where applicable.

- **Primary Goal 1:** Maintain fluency with basics from CSPB1300 by completing all Learning, Easy-, and Medium-level problems available in Exercism's Learning Mode for the topics Booleans, Namespaces, Strings, Includes statements, Numbers, Comparisons, Conditionals, and Loops (**23 total problems**).
 - **Subgoal 1:** Expand knowledge and improve coding style by reviewing community solutions and attending a weekly office hour to discuss questions.
- **Primary Goal 2:** Expand knowledge base by completing all Learning, Easy-, and Medium-level problems available in Exercism's Learning Mode for the topics Classes, Arrays and Vectors, Switch Statements, Enumerations, Headers, References, and Functions (**13 total problems**).
 - **Subgoal 1:** Build foundational knowledge by completing modules in [this introductory Udemy course](#) and reviewing a recommended textbook ([Sam's Teach Yourself C++](#)) as needed.
 - **Subgoal 2:** Expand knowledge and improve coding style by reviewing community solutions and attending a weekly office hour to discuss questions.

- **Stretch Goal (Dropped at mid-semester Project Update Report):** Complete all Medium- and Hard-level Exercism exercises in Practice Mode (**31 total problems**).
 - Consider increments of 5, 10, 15, ... , 35 problems completed to be milestones toward this stretch goal.

Materials and Methods

- (1) Exercism C++ track
 - (2) Udemy Course: Beginning C++ Programming – From Beginner to Beyond
 - (3) Sam’s Teach Yourself C++ in One Hour a Day
-

Results and Assessment

- **Primary Goal 1:** Fully met assessment goal; 100% of target exercises completed. Key learnings are described in more detail below:
 - **Closed a lot of knowledge gaps about modern C++ program structure and syntax.** At the beginning of the summer, I had never seen program structures comprised of multiple separate files like the ones used on Exercism and was quite confused; even debugging my exercises in my own IDE was a major challenge. Learning how to use different namespaces by leveraging the scope resolution operator (e.g. [Last Will](#)) and to correctly create and use modular header files with function-specific and .main() .cpp files to build, debug, and run a program was a big step forward in my understanding of how large-scale, maintainable software is actually built.
 - **Key exercises:** Essentially all of them!
 - **Significantly improved my proficiency with string parsing.** I initially expected to breeze through the Strings unit on Exercism, but it quickly became obvious that my “knowledge base” of C++ string methods contained a lot of Pythonic ideas that simply didn’t work. (I distinctly remember the moment when I realized that dictionary objects don’t exist in C++ while trying to solve Atbash Cipher. Oops!) This turned out to be an opportunity not just for review but for modernization. Our CSPB 1300 textbook used C++11 and earlier, so this was the point where I really started to integrate the learnings from my Udemy course and to modernize my approach to C++ syntax with concepts like initialization strings and modern C++ string methods (e.g. using string.at() instead of indexing for its inbuilt bounds-checking capability).
 - **Key exercises:** [Log Levels](#), [Atbash Cipher](#), [Rotational Cipher](#), [Protein Translation](#), [Reverse String](#), [Bob](#)
 - **Reviewed key number theory, counting, and cryptography concepts from CSPB 2824 and learned to implement them in C++.** Outside of the mid-semester Python cryptography project, I didn’t actually practice

implementing algorithms for a lot of the concepts that we covered in discrete math in any language. Discrete math is foundational to CS and repetition is the key to retention, so I was happy to find a lot of puzzles that tested and expanded my knowledge of modulo operations, counting formulae, and base conversion algorithms in the exercises. These exercises also provided a lot of opportunities to keep building on my progress with string methods above, and to work with new concepts like static casting and string-to-integer methods to create iterables from numeric arguments.

- **Key exercises:** [Raindrops](#), [Atbash Cipher](#), [Rotational Cipher](#), [Difference of Squares](#), [Trinary](#), [Luhn](#), [Hexadecimal](#), [Armstrong Numbers](#), [Eiliud's Eggs](#).
- **Primary Goal 2** Partially met assessment goal; 31% of exercises completed. Key learnings are described in more detail below:
 - **Completing additional coursework should have been an explicit milestone for this goal instead of a supplemental resource.** In addition to getting my first exposure to object-oriented programming, there were several concepts covered lightly in CSPB1300 that I needed to build more proficiency with, particularly vector operations, pointers, and references. Once I finished Main Goal 1, the majority of my weekly time allocation went to structured coursework in these topics, but I was able to complete a few exercises to reinforce these concepts. As always, they also both increased my C++ proficiency and reinforced concepts from earlier classes (e.g prime factorization, string parsing).
 - **Key Course Modules:** Arrays and Vectors, Pointers and References
 - **Key exercises:** [Making the Grade](#), [Prime Factors](#), [Paired Brackets](#)
 - **Working with object-oriented programming concepts for the first time.** My mental approach to programming is still extremely procedural, and learning to think in an object-oriented framework will be an ongoing learning process as I move forward into my Data Structures course this fall.
 - **Key Course Modules:** OOP – Classes and Objects
 - **Key exercises:** [Ellen's Alien Game](#)
- **Bonus Outcome:** Transferrable knowledge that synergized with other coursework.
 - **CSPB 4122: Familiarity with HTML shortened the learning curve on Dash.** Although I completed my LinkedIn Learning requirement by taking a short course on Plotly and Dash, one of the most useful pieces of knowledge transfer to complete my CSPB 4122 final project (implementing an interactive Dash app with Plotly visualizations) actually turned out to be fighting with formatting issues on my personal website in the early weeks of this course and fixing it by learning how to translate my formatting from

Markdown syntax into proper HTML tags. The learning curve to implement the required HTML objects to get my app running would have been significantly steeper without this early struggle.

Reflection and Conclusions

As I discussed my plans for the summer semester with my mentors, I spent a lot of time considering whether it made more sense to work on my C++ proficiency on my own time or as one of my Professional Development credit opportunities. I'm so glad that I decided to undertake this project as a class; although it was predominately solitary, independent research and practice, I gained access to resources that I was previously unaware of and benefitted from the feedback and structure of a formal course with weekly check-ins. The short summer semester imparts an added intensity to deadlines, and the temptation to shelve my C++ practice once my other classes hit peak intensity in July and early August would have been overwhelming if I wasn't accountable for dedicating 4 hours of my week to my project goals.

If I had to sum up what I accomplished in this project with a single phrase, I would classify it as "bridging and connection". Although I didn't solve as many problems as I initially hoped to (and realistically should have focused on quality over quantity from the get-go, as my peer reviewers wisely pointed out), I think I did successfully retain and build on my learnings from all of my spring coursework and succeeded in clarifying some of the unknowns of structures and object-oriented programming that would have steepened my learning curve in the early weeks of CSPB 2770.

When I completed my first degree as a traditional undergraduate, I was much less cognizant of time than I am now. Even so, I still underestimate how long the learning process will actually take when I'm building foundational knowledge in an unfamiliar domain. For example, Main Goal 1 took much longer than I anticipated based on the difficulty of the first 10 exercises; I originally targeted June 23rd but didn't successfully complete all exercises until July 18th, and I wasn't able to fully complete Main Goal 2 as a result. I've learned that implementation is often the easy part; the biggest time sink is also the hardest thing to forecast: discovering the right keywords and combination of resources to turn an "unknown unknown" into a "known unknown" requires a foundational of knowledge sufficient to quickly narrow the search space. One way that I could become more efficient at this process is to "fail fast"; I need to recognize when I'm unproductively blocked and become less reticent to seek out mentors who can help me narrow my search space more efficiently. Although the biggest blocker to finishing more of Main Goal 2 was being unable to allocate additional time beyond the core block to studying, spending more time in office hours or using Exercism's asynchronous mentoring service more frequently might have enabled me to complete a few more exercises for Main Goal 2. I'll keep looking for opportunities to grow in this area in the future!

Links to Project Deliverables

- [Personal Website](#)
- [My solutions repo](#)
 - This is also where my Final Report is stored
 - Also: see links to Exercism publication for solutions above