

## **Narrative - Enhancement One: Software design and engineering.**

The artifact chosen for this first category comes from the course CS-300 "Data Structures and Algorithms Analysis and Design," it was created in April 2024 and is a software to manage the university courses and prerequisites to be used as a helpful tool for a student advisor to register the student to the various courses. This artifact was chosen as it has been evaluated as a perfect starting point to apply numerous skills to bring it to a more realistic product that can be potentially employed in a production environment. The first improvement applied is to transition it to a more "modern" language like Python and use its characteristics to create a safer product. It has given me the occasion to show coding skills, especially in clean and well-written code that is easily understandable by applying the Python programming style guide. Another skill showcased is the ability to transition to a different data structure paradigm from standard C++ vectors to more modern lists and sets; moreover, risky pointers have been moved to safer objects.

The first planned outcomes were to design, develop, and deliver a professional-quality product, and even a bit overkill by adding detailed comments to each part of the code so everyone who reviews the code will gain an understanding of its logic immediately. Another outcome met is the design and evaluation of computing solutions that solve a problem or improvement. It was achieved by rewriting everything and optimizing the code by observing algorithmic principles, computer science practices, and standards. Finally, a security mindset was developed by paying greater attention to string sanitization while loading the course data from the CSV file.

While enhancing the artifact, I've learned how adding comments to uncommented parts of code can let you better reflect on the algorithm or process used to achieve a solution and consider different paths that could be more efficient and better aligned to security. Another personal improvement is becoming more used to modern data structures like lists and sets and understanding how helpful it can be to use the module's methods dedicated to these new datatypes, mainly to avoid design flaws leveraging the language automation already in these structures. During this journey,

one of the biggest challenges faced was to adapt and being able to understand some new constructs for cycles that traverse lists of objects that are perfect for readability but sometimes hide the working principle by compressing the instructions in a single line of code, for example, when loading all the course numbers in a set with the row `"all_course_numbers = {crs.number for crs in courses}"`, when every single line of the CSV file is analyzed with a for a statement like: `"for row in reader:"` or extracting the prerequisites with `"prerequisites=[field.strip() for field in row[2:] if field.strip()]"`.