The original project was the computer science department at ABCU needed a code where a student code finds the classes that they are taking. This needed to be able to load a file from a data structure. Print an alphanumeric list of all the courses in the Computer Science department. Print the course title and the prerequisites for any individual course and exit the program.

The primary enhancement involved upgrading how prerequisites are handled. Originally, each course stored prerequisites as a single comma-separated string, which limited flexibility and made it hard to manage multiple dependencies. I updated the code to store prerequisites using a vector<string>, allowing for any number of prerequisites per course. To support this, the file loading function was rewritten to use a string stream to split each line of the course file into tokens. The first two tokens are used for the course number and title, while any remaining tokens are collected as individual prerequisites. This makes the data structure more robust and reflective of real-world course catalogs.

I added logic for recursively displaying the full chain of prerequisites for any course. When a user selects a course, the program now prints not only the course's immediate prerequisites but also walks through each prerequisite’s own dependencies in a nested, indented format. This helps users visualize the full path of required coursework leading up to their selected class. A helper function (printCourseChain) performs this recursive traversal, ensuring that all prerequisite levels are clearly shown in order.

Memory management was also addressed by introducing a destructor and a recursive clearTree function to properly deallocate dynamically allocated memory used by the binary search tree. This prevents memory leaks and ensures cleaner shutdown of the application. The input handling logic was also slightly improved for clarity and user feedback. An example would be printing a confirmation when course data is successfully loaded.

Together, these changes transformed the code from a basic course lookup tool into a functional course planning assistant, capable of guiding users through their academic path with prerequisite chains clearly mapped out.