// Keaton Spiller

// CS302

// 03/20/2024 Winter 2024

// Program # 4/5

Efficiency Writeup

Throughout this assignment I learned about the difficulties of pass by value programming languages when building my own data structure. In C++ I took for granted the simplicity their classes provide, and freedom that is granted when I can pass by reference. In this python assignment I created a BST of dictionary objects {key : value} / hash map classes of the different characters. I made sure to that the Node didn’t inherit the class, and the BST didn’t inherit from the Node similar to the other assignments showing what I learned throughout the process.

I created black box tests that followed my hierarchy of UML diagrams, and this was really helpful to figure out my plan of action. Before drawing up the UML I had no idea what direction I was going to take, and this assignment solidified for me that I’m a conceptual / visual thinker who has a harder time understanding the concepts without drawing them down onto paper. Writing out concepts onto a white board, drawing out flow diagrams, and sketching out my ideas is one of my strong suits, and this class gave me a deeper understand on how to take this to the next level.

I didn’t get to all the test cases that I would’ve like to finish, actually after I ended up writing the glass box implementations of most of my functions I ran out of time, and wasn’t able to fix the tests cases that become broken after implementing my design. Many times my initial test cases were asserting if the values were None or False Boolean expressions, but as I built out the programs, return values would change depending on what was necessary to the get the functionality implemented.

I didn’t get the excepting handling that I wanted to accomplish for this assignment. I would’ve liked to create a list of exceptions that I could loop through and catch for all the different variations of potential errors that could arise in a program. I had created a combination of black box and glass box tests, but with the newest changes to the program, many of my unit test cases no longer worked.

Usually when I has used exception handling in the past I would have a try catch block individually for every scenario, yet the knowledge to use a struct like list, and automate the throw / raise of multiple different errors that could occur in one try block to catch and account for the variations of intermittent bugs is something that will help me reach the next level of programming in the future.

Overall, I’m happy with how far I’ve come in two quarters, learning advanced C++ programming when strictly knowing python was something I structurally needed as an individual. I now feel confident in being able to learn any programming language in roughly a month and will continue to improve on my skills especially in exception handling, glass box/ black box unit testing, integrations testing, parallel and concurrent programming, and explaining my code.

Even though this last assignment wasn’t as polished as my other assignments, it’s not always about where you end up but the journey that you explored along the way. Throughout this class I’ve learned how to code in any language and built a structural foundation that will help me in anything I venture towards in the future. I put in the overtime over the weekends, read the textbook thoroughly, attended every lab, and didn’t miss a single class. I really invested myself into learning these tools, and now I feel confident in approaching new challenges. using these high-level ideas I feel that I’ve learned how to learn, and discover the patterns that exist amongst the sea of information.