Kyle Zalewski

CS202

Efficiency Writeup

This project gave me a lot more practice with working in a hierarchy, and allowed me to continue to hone my skills in creating these structures of class objects. I used RTTI to organize my interview questions into the various categories, which were held in respective elements of the overall array of doubly linked lists. Each list itself contained several questions, which had data for their type intrinsic to the type of pointer they were represented by. They also contained a difficulty rating, which was passed in as an argument while building the array.

One thing I think this project did well for me was to help me refine my process for building the test data structure. In previous projects, I’ve started at the Node level and built out from there, constructing many objects in my main test code that would later have to be deleted. With this project, I created only a handful of objects, and passed in arguments that allowed all of the information through the hierarchy to be filled in as the various constructors and helper functions were called. In this way I was able to streamline my testing process, and spend less time creating data structures and more time correcting the code in those classes.

One thing that I have yet to perfect is the use of destructors. The destructor in my array class was not able to successfully clear out all the allocations within its lists, which ultimately ended up being the Node destructor’s inability to properly delete its question pointer. I will need to continue investigating the cause of this error to clear out all possible memory leaks. Luckily, that was the only memory leak remaining at the conclusion of the coding process.

RTTI seemed a bit clumsy as a method for organizing the questions into the array. While it’s nice to have the dynamic type identification handling things in a safe (and fairly foolproof) manner, if this were a production level software, I don’t think this would be the best method. If something gets completely miscategorized it seems like you could trap certain pointers in part of the hierarchy they don’t belong and the only option to recategorize them would be to destroy them completely and rebuild. I’ll have to continue working on my RTTI and dynamic cast skills in order to confirm that theory. For this project, it seemed to work very well.