Juhwan Lee
Professor Karla Fant
CS202
November 22nd 2020

Program #4 & #5 Design Write-up

With both programs #4 and #5 we will be implementing our solutions using java. Our goal must be to develop an object-oriented solution but this time we will implement it in java. In these last two programs, we will implement two data structures: array of linear linked lists and a balanced tree. In program 4, we will be implementing a playlist as an array of linear linked lists and in program 5, we will be implementing a balanced tree where each node has a playlist of items. This programming assignment is to design a system that would allow media to be managed by our software once uploaded. Example operations are uploading media, removing media, publishing it for general viewing, or publishing it for a specific class. Each video can have comments, attachments, and descriptions. Moreover, authors can look at the analytics of how often these videos have been watched (or re-watched). In addition, quiz questions can be added to the media. Each course can have multiple playlists of these media available for the viewers to step through. In program 4, we are to build the part of the application that creates one playlist. In program 5, we will allow for classes to be supported each with multiple playlists. Our job is to come up with a design of an OO framework that will support an application. The key is to make sure to solve this problem using Object Oriented methodologies with dynamic binding and function overloading. For these two programs, the use of external data file(s) will be necessary because we will be working with lots of datas. First of all, my overall design is as follows. The media class is set as a common base class, and the general class, cs162 class, cs163 class, and cs202 class are inherited from the media class. The data members of the media class are name, comments, attachments, description, and views. And the data members of inherited classes are quiz, answer, and student-answer. And node class has media data and next pointer as data members, and uses setter and getter functions. And the playlist class has a head node as a data member and has functions such as publish_general, publish_specific, remove_general, remove_specific, etc. in the playlist class. Therefore, in the case of program 4, if the user is a professor, functions such as uploading media and deleting media can be performed, and if the user is a student, functions such as accessing media and taking quizzes can be performed. And the playlist consists of an array of linear linked lists, and each array is divided into general, cs162, cs163, and cs202 to form a different linear linked list. And in the case of program 5, the tree data structure is added to program 4. The tree data structure consists of a node class and a system class. Because we are using one more node class, I changed node to node1 and node2 so an array of linear linked lists can use node1 and a balanced tree can use node2. The node2 class has playlist data, left pointer and right pointer as data members, and the system class has a root node2 to store various playlists in a tree structure. In the case of program 5, since there are many playlists, the name data member was added to the playlist class, and there will be many playlists other than the computer science playlist, so instead of the cs162, cs163, and cs202 classes, the derived classes will be class1, class2, and class3. While programming, the overall design may change slightly, but for now, the overall design is as described so far.