Rationale and Change Reflections

ICT283 – Data Structures and Abstractions – Assignment 2

32067232 – Jake Kroon

Rationale

For this assignment I decided to use map for students and a tree for units. My idea for this was that a student can be easily paired to a student Id the key for the map. Having a consistent theoretical complexity this would be helpful as you have many students. Also a map provides the ability to iterate, I feel that iterating through a map of students is the most likely rare case scenario for all but still a necessary feature to have. I felt a multimap was unnecessary and if I put students in my tree I would have to heavily modify it making it no longer as “minimal” as it could be. Also with students there is a very low chance for data duplication assuming this institution understands the idea of making student identification number unique, so I do not have to worry about removing duplicates which is a side effect of my binary search tree.

I used the tree for units as I felt that that in the units there would exist the most duplicates as many students can take the same units. Using the tree I can eliminate all these duplicates reducing the amount of memory I have to use considerably. I made my binary tree an AVL tree as I feel that in a large institution in the future my tree could contain many units and though there is added processing time initially, if this became a persistent system and there were lots of units already in the tree. It would be a rare occasion that more were added and then search time would be the most important factor. Having a perfectly balanced tree I can take advantage of the main benefit of a tree, avoiding having a link list by accident. Units in the tree are compared by their unit Id ASCII values using the C++ compare function. Though input my same incorrect when viewing the traversals, it is actually correct because of the way integers behave in strings due to their ASCII values. Also as the search and insertion comparisons are the same, this is a negligible effect on processing time.

Reflections

I made quite a few changes to my overall design from the first assignment I will list them as dot points with a brief description below

- Unit class moved to main as an object contained in the BinaryTree
  - Due to this change I had to remove all the set and get methods for it through the hierarchy and add the unit id to results so as to have a “foreign key” of sorts to keep access when looking up units through the tree.
  - I did this so I would not have duplication existing in my program.
- StudentIO no longer contains the primary data structures
  - StudentIO is now a IO class that is an aid to the client program dealing with input and output. This is to keep the main program tidy. Through having the primary data structures in main the client can now extend their program as they wish, whereas before they were hidden and the client could not extend their program.
  - A method was added that removes all newline characters as well as leading and trailing white space from a string to ensure the integrity of data going into the data structures.
  - Output had to be changed to handle the new data structures
- Student
  - New GPA calculation methods had to be created to meet the new requirements
  - GPA parameters had to change to handle the new data structures
  - Function pointers had to be added as the student class is the primary class that needs to access the map.
  - A static Unit object was added with appropriate static methods to provide a way of setting the object for access when calculations needed to be performed i.e. the GPA.

- Date
  - I changed the month to an integer instead of a string
  - A string can be obtained that has the date in a dd/mm/yyyy format