Project Interim Report

Group Number: P18

Group Members:
Bryan Wong
Jad Alriyabi
Avash Singh Thapa
Gurnek Ghatarora

Date of Submission: 11/2/2020

System Design and Implementation

Student class implementation:

The Student class holds information about the student like their CGPA, research score, first name and last name. We maintain the class as an ADT by keeping its data variables as protected and use get and set functions to receive and manipulate the class' data. We decided to keep the student's data as protected so that we can access the variables in our child classes. We added comparison functions to the class to compare the student's CGPA, research score, first name or last name. These compare functions are needed later on in our derived class' sort functions.

DomesticStudent class implementation:

This class is a derived class of the Student class which means it is able to access all the public functions and protected variables of the Student class. It maintains ADT by keeping the data variables of Student and it's own as private. This class has one member function to compare a domestic student's home province. This compare function is later used in the overall sort function. We also implemented various sorting functions to sort an array of DomesticStudent objects by CGPA, Research Score, First Name, Last Name and Overall. Lastly, we have a print function to output all the information in a DomesticStudent array.

InternationalStudent class implementation:

This is another child class of the Student class. We create this class using inheritance in order to not only keep our program clean and easy to follow, but also to maintain certain variables which we want present in all classes. We keep the characteristics of ADT here once again, by making all the private variables represent data variables. Like our DomesticStudent class, we have overloaded the insertion (<<) operator to print out the domestic student's data. Other public member functions include a compare function for country, individual sorting functions for CGPA, Research Score, First/Last Names, and an Overall sort. Lastly, we have a print function to output all the information in an InternationalStudent array.

Toefl class implementation:

The Toefl class holds data about a student’s Toefl scores. These scores are private variables so we implemented mutator functions to access and manipulate them. We added a “passed()” function so we can determine if the student has passed their Toefl exam. We’ve also overloaded the insertion operator to print out the scores in each section and the total Toefl score.

Main function implementation:

Our main function makes use of file IO to read a .txt file containing each student’s full name, CGPA, research score, country/province and Toefl score if they are an international student. After a successful read, we use the set functions to save the student’s information into an array of its respective class type (domestic or international). Then our program outputs a menu where the user is able to make use of the various sort functions we have implemented. We made use of the overloaded insertion (<<) operator to also print the array and related information to the console every time a sorting function is called.

Testing Plan

Our testing plan involved individually testing each function to make sure that they all accomplished what we wanted it to.

For example, there were various problems in the .txt files that we ran into when we were sorting. While sorting by first name, we realized there were different ways the names could have been input. Two cases such as ROHIT and tanvi were different and would have had to be sorted differently because of how the ASCII table holds the letters. This caused these names to be incorrectly sorted. Another example is in CGPA, if there was an input of 4.34, it would be rounded down to 4.3 gpa and deemed as a legal input. These were just some errors we found when we were debugging in the end.

Task Distribution

We tried to distribute our tasks very evenly. We all had our midterms and other commitments for most of the time this project was out so we sparsely met. But whenever we did talk, we came out with very fruitful results. With 2 full days of collaborations, we got most of the work done in the final weekend before the project was due. Other times we were off researching how to implement the sorting function.

Weekly Project Plan

Names	Week 1	Week 2	Week 3	Week 4
Jad Alriyabi	Finish Lab 3	Discuss task distribution	Sort function for first name and last name	Project Report
Bryan Wong	Finish Lab 3	Discuss task distribution	Sort function for first name and last name Class definitions	Overall sort Main function Debugging
Avash Thapa	Finish Lab 3	Discuss task distribution	Sort function for CGPA and research score	Overall sort Main function Debugging
Gurnek Ghatarora	Finish Lab 3	Discuss task distribution	Sort function for CGPA and research score	Overall sort Debugging Testing