# Part 1

1. <Insert Picture of design>
   1. Need to figure out how students will be stored

## Requirements:

### Display all the information related to this institution in O(1).

* use simple getter function

### Display the number of students transferring to this institution in O(1).

* Store number in private section of Institution object
* Use simple getter function

### Display the number of students transferring from this institution in O(1).

* Store number in private section of Institution object
* Use simple getter function

### Modify all information related to this institution (except the "key(s)" used for "institution") in O(1). This requirement does not include the modification of the content of each student found in either student listings of this institution.

* Use simple setter functions to access all private members
* Does not modify the Student objects held in a data collection

### Display all students transferring to this institution, by ascending alphabetical sort order of last names, in O(n) where n is the number of students transferring to this institution.

* O(n) implies traversing every node
* Data must be sorted by ascending alphabetical order of last names
* Can used an AVL tree to sort by last name

### Display all students transferring from this institution, by descending numerical sort order of student numbers, in O(m) where m is the number of students transferring from this institution.

* O(m) implies traversing every node
* Data must be sorted in descending numerical order of student numbers
* Use an AVL tree to sort by student numbers

### Insert a student transferring to this institution in O(log2 n), where n is the number of students transferring to this institution.

* Insertion of Log2n achieved by a using a binary tree

### Insert a student transferring from this institution in O(log2 m), where m is the number of students transferring from this institution.

* Insertion of Log2n achieved by a using a binary tree

Modify all information related to this student transferring to this institution (except the "key(s)" used for "student") in O(1).

* Use simple setters to modify the student object information
* Use 1-1 hashing to find student object

### Display all information related to a student transferring to this institution in O(1).

* Use simple getters to get the student object information
* Use 1-1 hashing to find student object

Modify all information related to this student transferring from this institution (except the "key(s)" used for "student") in O(1).

* Use simple setters to modify the student object information
* Use 1-1 hashing to find student object

Display all information related to a student transferring from this institution in O(1).

* Use simple getters to get the student object information
* Use 1-1 hashing to find student object

1. Header files of “student” and “institution” objects