# Part 1 – Design

1.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 0 | 1 | 2 | 3 | 4 | … | n+m |  |  |
| 0 | name: SFU  **studentTable**  **\*TOPtr**  **\*FROMPtr** |  |  |  |  |  |  |  |  |  | Hash table containing pointers to all student objects in this institution |
| 1 |  |  |  |  |  | lastName: Lui  firstName: Curtis  studentNumhome: 12345678 |  |  |  |  | AVL tree containing all student node objects transferring TO this institution. Implemented using nodes and links and is sorted by lastName. Able to display students in ascending order by last name by in order traversal |
| … |  | lastName: Lee  firstName: Bobby  studentNumhome: 45579890 |  |  |  | lastName: Snow  firstName: John  studentNumhome: 98643723 |  |  |  |
| 1000 |  |  |  |  |  |  |  |  |  |
|  | Hash table containing institution objects that the user can select from a listing of institutions displayed on the application's window. Implemented in an array | lastName: Le  firstName: TJ  studentNumhome: 11677896 |  |  |  |  | lastName: Ross  firstName: Bob  studentNumhome: 33225511 |  | lastName: Jerry  firstName: Tom  studentNumhome: 55778899 |  | AVL tree containing all student node objects transfering FROM this institution. Implemented using nodes and links and is sorted by student number |