

# HW8: Ski School

---

*CSS 501 – Data Structures & Object-Oriented Programming*  
*By: Hansel Ong*

## Summary

For this homework, you will simulate basic data structure management for a hypothetical ski school. Assume a ski school can have an indefinite number of instructors and students. In an “ideal” scenario, each ski instructor should have no more than 5 students. Each student is assigned to an instructor on an “as available” basis (i.e. whoever is most available). However, students can also request a specific instructor. When all instructors already have 5 students, priority should be on maximizing the number of instructor with only 5 students, with other instructors having up to 8 students. When all instructors already have 8 students, any additional students will be assigned to instructors with the least number of students. Create a tree data structure with inserts based on these requirements.

## “Formula”

- Less than 5 students per instructor: Students distributed evenly across all instructors -> 5 instructors and 5 students means each instructor has 1 student
- More than 5 students, up to 8 students per instructor: Students should be added to an instructor until the instructor has 8 students before adding students to another instructor -> 3 instructors and 20 students means one instructor has 8 students, another instructor has 7 students, and the last instructor has 5 students
- More than 8 students per instructor: Students are distributed evenly across all instructors -> 3 instructors and 28 students means two instructors has 9 students and one instructor has 10 students.

## Assignment Description

Create a tree data structure used to keep track of the above data. Implement insert and remove for the tree data structure (what should happen when an instructor is “removed”?)

## Grading Criteria

- Implementation
  - [2 Points] Tree data structure (“root” will be one “supervisor” overseeing all instructors)
    - [1 Point] keep track of “Person” object – representing student/instructor
    - [2 Points] keep track of “Student” and “Instructor” objects – where both “Student” and “Instructor” objects derive from a “Person” object (doing this means that your tree is also keeping track of “Person” object so you get the above point, too)
  - Insert and Remove
    - [5 Points] Follows the above formula/invariant
    - [2 Points] Insert instructor
    - [2 Points] Insert a student without specifying instructor
    - [3 Points] Insert a student with specifying instructor
    - [2 Points] Remove student
    - [2 Points] Remove Instructor
  - “Find” a teacher
    - [2 Points] Given the teacher’s name
    - [2 Points] Given the student’s name
  - [3 Points] “Display” function – “prints” out all teachers with all students under each teacher, e.g.
    - Teacher 1
      - Student 1
      - Student 2
      - Student 3
    - Teacher 2
      - Student 1
      - Student 2
- Demonstration
  - A “ski school” containing each of the following
    - [1 Point] 3 teachers, 19 students (create using “inserts” and “display” this)
    - [1 Point] Remove one teacher from the previous tree and “display” (there should be 2 teachers, 19 students remaining)
    - [1 Point] Attempt to insert a student for the teacher that was previously removed (what should happen?)