Build an elevator for an office building.

10 people, 1 elevator, 5 floors.

Elevator has a weight limit of 700 pounds.

Weight on elevator can’t exceed limit.

Elevator moves up and down at a rate of 1 floor per half second.

People have a weight between 100 - 250 assigned randomly.

A Person’s name should be Person1 - Person10.

People all start in the lobby on floor 1 and they all arrive in the order of their assigned number and wait in the lobby for their turn to ride the elevator to the floor they work on.

People will need to go up to the floor they work on, assigned randomly to each person (one person should be on each floor though).

A Person will stay for 3 seconds on their floor (the length of their workday) at which point they will be waiting in a line at the elevator to go back down to the lobby on the first floor so they can go home (who wants to stay at work longer than is required?).

The path your elevator takes should be to the closest floor that is requesting a pickup or that a passenger needs to get off on in whatever direction it is heading. If the elevator is heading up and has no more passengers that need to go to a higher floor and no requests from a higher floor it should change directions. The same is true in the other direction.

The elevator should handle all requests as it passes a floor (keep in mind the weight limit). Everyone in the building is polite though, so they will only get on the elevator if it is going in the same direction that they need to go i.e.) up or down.

The priority of who gets on the elevator at a given floor is based on first come first serve.

You may not use any data structures from the collection framework in your implementation, but you may use your own custom data structure such as those made in previous weeks.

Your solution does not need to be optimized, but you should understand where you might be able to make modifications to improve your implementation and mention them in Assignment 3.

You should work on Assignment3 first, design will be key. This can be a lot of work, or it can be a little (well maybe not a little) work. It will depend on how well you design your solution.

**Grading:**   
Correctness: You can lose up to 20% if your solution is not correct   
Quality: You can lose up to 20% if your solution is poorly designed   
Testing: You can lose up to 20% if your solution is not well tested   
Explanation: You can lose up to 40% if you cannot explain your solution during the grading session