Programming Project 10

This assignment is worth 70 points and must be **completed and turned in before 11:59 on Monday, December 7th 2009.**

Assignment Overview
This assignment will give you more experience on the use of:
1. classes
2. class methods
3. object-orientated programming

The goal of this project is to gain more practice with classes and Object-Orientated Programming (OOP). OOP is a very popular programming paradigm and supported by Python very well. You can refer to wiki to get more details.

In this project, we are going to simulate a simple elevator. You will also show how different strategies can affect the efficiency of an elevator.

Task
Your task is to implement the simple elevator in Python using classes. Before implementation, please play the demo. It would help you understand the project. We would like you to implement your own "efficiency" strategy for dealing with customers. The default strategy is the simple "start at the bottom, go to the top, then go to the bottom". Can you write a better strategy, one that is more efficient?

Project Description / Specification

1. Create three classes: Building, Elevator, and Customer.
2. Equip the building with an elevator. Ask user to customize the number of floors and the number of customers.
3. Program should have error checking to make sure the user inputs are valid. For example, if a user gives non-integer inputs, notify the user that the inputs are incorrect and prompt again.
4. Each customer starts from a random floor, and has a random destination floor.
5. Each customer will use the elevator only once, i.e., when a customer moves out of the elevator, he/she will never use it again.
6. When all customers have reached their destination floor, the simulation is finished.
7. Part of the grade on this project will be the appropriateness of your classes, methods, and any functions you use. The quality of the code will now matter as well as the performance. No skeleton file is provided; you need to come up with this yourself.
8. All classes' methods require a docstring for a general description of the method.
9. Implement both your own strategy and the default strategy and compare. Your strategy does not have to be better (your grade will not depend on it) but the comparison is required.
10. Don't use any global variables.

Deliverables
proj10.py -- your source code solution (remember to include your section, the date, project number and comments).

1. Please be sure to use the specified file name, i.e. "**proj10.py**"

2. Save a copy of your file in your CS account disk space (H drive on CS computers).
3. Electronically submit a copy of the file.

Notes and Hints:
Here are some suggested attributes and methods. Note that you needn't follow these faithfully.

Class	Attribute/Method	Explanation
Building	num_of_floors	The number of floors
	customer_list	The list of customers
	elevator	The elevator equipped in the building
	run(self)	The method to operate the elevator
	output(self)	Output the building (check the demo program)

Class	Attribute/Method	Explanation
Elevator	num_of_floors	The number of floors
	register_list	The list of customers in the elevator
	cur_floor	The current floor of the elevator
	direction	The direction of the elevator
	move(self)	The method to move the elevator by 1 floor
	register_customer(self, customer)	A customer goes into the elevator
	cancel_customer(self, customer)	A customer goes out of the elevator

Class	Attribute/Method	Explanation
Customer	cur_floor	The current floor of the elevator
	dst_floor	The destination floor of the elevator
	ID	The customer's ID
	in_elevator	The flag to denote whether he/she is in the elevator
	finished	The flag to denote whether he/she has reached dst_floor

Notes and Hints
In your main function, at the beginning ask the user for the number of floors and the number of customers which can be used to create an instance of Building. Then we only need to call the run() method and output() method repeatedly in a while loop.

Randomly select the floors (to and from) for each customer. Use the randint function from the random module (look it up in the Python documentation)

To compare efficiency of strategy, count the number of floors visited for your strategy versus the default strategy

Use the demo to create useable output from your simulation.