CST8219 – C++ Programming

Final Project due December 10, 11:59 pm

This project description is a first draft. You will certainly have questions and there will certainly be issues that we haven’t thought of, which is how things work in the real world. The professor will post a new version of the document every week. The requirements will not change, however clarification on how the requirements are to be implemented might be added.

For this project, you will create a car reservation program. Your system should track customers, rental locations, and vehicles for rent. There must be a graphical user interface written using the C++ Nana library. Using any other software library will be assigned a mark of 0 automatically. All work is to be done individually, and you may not share your code with anyone else. You can also not accept any code from anyone else. As a reminder, here is Algonquin College’s definition of cheating, plagiarism, and various other things you should not do: <https://www.algonquincollege.com/policies/files/2021/09/AA48.pdf>

Requirements:

1. The user must be able to add and remove customers, add and remove rental locations, and add and remove vehicles. (3 marks)
2. Vehicles can be assigned to a rental location where they are picked up and returned. A vehicle can also have no location assigned, meaning that they are not for currently available for rent. (1 mark)
3. Customers can be added and removed from the system. You should store their first and last name, street address, and age. Customers can also be designated “preferred customers” meaning that they get 20% off the price of any car rental. Customers that are 25 years or younger must pay 20% extra for their rental, and cannot be designated preferred customers. (2 marks).
4. Customers can rent a car from a rental location for a number of days. They must specify the day that the rental begins, and the expected day to return the car (1). That makes the car object unavailable for rent by any other customers during that time(1). If wants to rent a car during a day that it’s already booked by another customer, then the system will tell them that the car is unavailable on the day that the other booking is starting(1). (3 marks)
5. The process of renting a car should follow this pattern:
   1. Select a username from a drop-down list. Select a rental location from a second drop-down list. A third drop-down list should them be populated with the cars that are available for rent at that location where the user can select the car they want to rent from the list.
   2. There should be a text field where the user can enter the starting date and end date of the rental.
   3. For the purposes of this assignment, when you start your program, the system starts at day #0. Then days are just numbered sequentially after that. Don’t keep track of names of days, or weeks or months. To keep things simple, just pretend that the days are just numbers that go up forever, so if you run this program long enough, you might reach day #345923 for example.
   4. There should be a button on the GUI to simulate advancing to the next day. When the user clicks this button, there should be a day number showing somewhere on the interface. (1 mark)
6. The user can also return a car, however the process is much simpler. The user should be able to view a list of all customers that are currently renting a car. When a customer is selected, the user can see which car they are currently renting, the number of days they have rented it for, and the total price of the rental period. There should be a button for returning the car on the current day that is showing in the interface. That should then make the car available for rent on the next day (a car cannot be rented on the same day that it is rented to simulated taking time to clean it and do maintenance). Also, if the car is returned past the expected return date, then they must pay 100% extra for the days that the car is late.
7. There should be a way to view the following information:
   1. All customers in the system. (1 mark)
   2. All rental locations in the system. (1 mark)
   3. All cars in the system. (1 mark)
   4. All cars that are currently being rented. (1 mark)
   5. A list of rentals that are late (cars that have not been returned by a customer by the expected return date). (1 mark)
   6. All customers that are currently renting a car. (It is probably best to use some sort of “rental object” that tracks which customer, which location, which car, the starting date, and the expected return date) (1 mark)
   7. The amount of money a customer owes your company (don’t worry about paying the amounts in this project, meaning that the amount will only ever get bigger as they rent more cars). (1 mark)