GLOBAL UNIVERSITY

Fall 2024-2025

CSC301 Data Structures and Algorithms

Project

Car Insurance System

Project Info:

The deadline for this project submission is ***Sunday January 26th at 11:59 pm***.

Submission Rules:

When starting out with your project, please create a folder on your laptop and include all files in that folder. When submitting your project, compress your folder and submit your zip folder. Please do not submit text files or screenshots of your code. Before submitting your project, make sure your code is running without any errors, and also provide a “Read me” text file that includes the names of the group members of your team alongside what each member worked on as part of the project.

Project Description:

In this project, you will be designing and implementing a system that manages the data of a car insurance company. This system will assist in managing **four** main things: clients, cars of clients, car accident claim requests, and car accident claims processed.

Clients:

Your system should be capable of storing all clients the company works with. For each client the system should store the following data: client name, address, license number.

The system will be storing the clients in a **Binary Search Tree** data structure. The name of the client **(in lowercase)** will be used as the key in the Binary Search Tree to properly place the client nodes correctly.

Each client has a list of cars associated with them.

Cars:

The insurance company provides services for cars that have been in an accident. The system should be able to store information about all of these cars. Each car should be linked back to its owner (client), so we should store the cars that belong to each client in a **Linked List** data structure connected to each client node.

A client can have many cars.

For each car the system will be storing the following data: license plate, model, year.

Car Accident Claim Requests:

The insurance company deals with accident reports that occurred for each car that is serviced by them. Each car has multiple accident claim requests that are linked to that car.

A car can have many accident claim requests. These requests are waiting to be processed by the insurance company. It is mandatory to process the claim requests in the correct order from when the accidents occurred. That is why the system will use a **Queue** data structure to handle the list of accident claim requests.

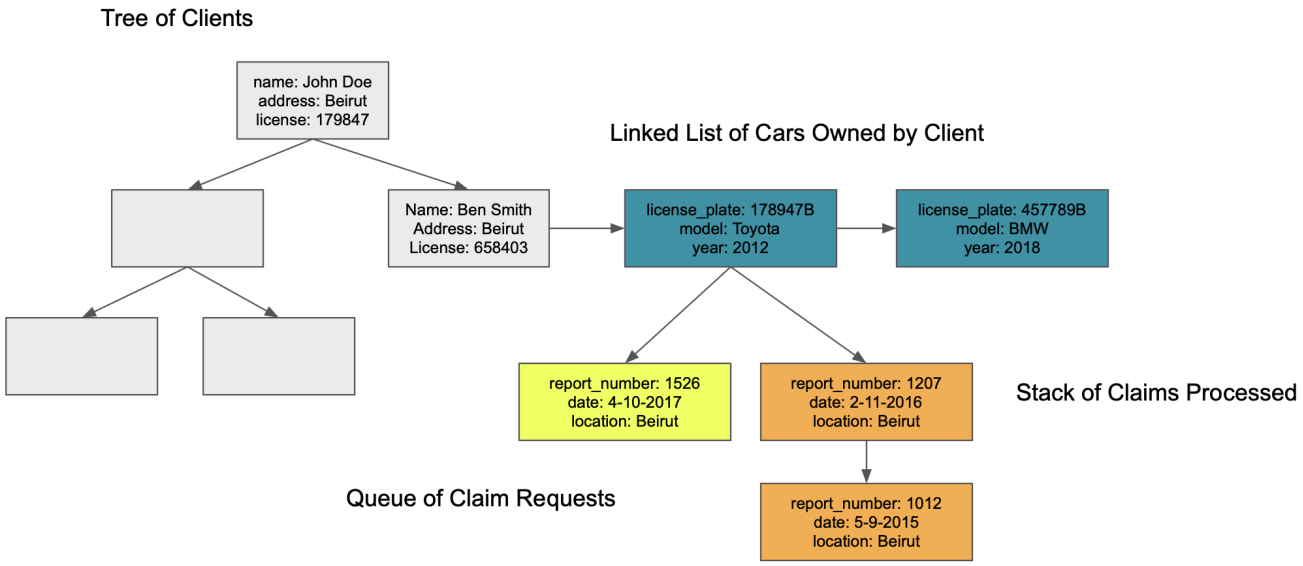
Each claim request is linked to a car and the system should store the following data for each claim request: report number, date, location.

Car Accident Claims Processed:

The company will go through the queue of accident claim requests for each car and process the

claims in order. We need to have a **Stack** data structure that is also linked to the car and stores the claims that have been processed. Each claim that has been processed gets dequeued from the claim requests queue and gets added to the claims processed stack.

The system will look something like this:



In this example, Ben Smith has two cars, had three accidents on one car (Toyota), where two of these accident claims got processed and one claim is still pending (requested). Notice that the claims that have been processed in order (based on date, which is how they are being placed in the requests queue).

Project Interactive Terminal:

In your main file, you need to write code to be able to show a list of commands to pick an action and then you run a function based on the action chosen. When you run your code, a list of commands should show in the terminal to pick an action to do. The first thing to show in the terminal are these commands:

1. Clients
2. Cars
3. Claims

If ‘Clients’ was chosen, you show the following options:

1. Add Client
2. Delete Client
3. Edit Client data
4. Print all Clients

If ‘Cars’ was chosen, you show the following options:

1. Add Car to Client
2. Delete Car from Client
3. Edit Car data
4. Print all cars of Client
5. Print all cars at company

If ‘Claim Requests’ was chosen, you show the following options:

1. Add Claim Request to Car
2. Process Claim Request from Car
3. Print all pending Claim Requests linked to Car
4. Print all Claims Processed linked to Car

Scenario:

You have been provided with txt files of sample data to import into your data structures before running any commands.

You are provided with: “clients.txt”, “cars.txt”, “claimRequests.txt”, “claimsProcessed.txt”

You will use the “clients.txt”, “cars.txt”, “claimRequests.txt” files to populate the data, and use the “claimsProcessed.txt” file to process claims from the claim requests.

You can add more data yourselves to test this more.

Grading:

- Build clients tree structure, its functions, and import “clients.txt” data (20%)
- Build cars linked list, its functions, and import “cars.txt” data (20%)
- Build claim requests queue, its functions, and import “claimRequests.txt” data (20%)
- Build claims processed stack, its functions, and import “claimRequests.txt” data (20%)
- Main file code and interactive terminal (20%)