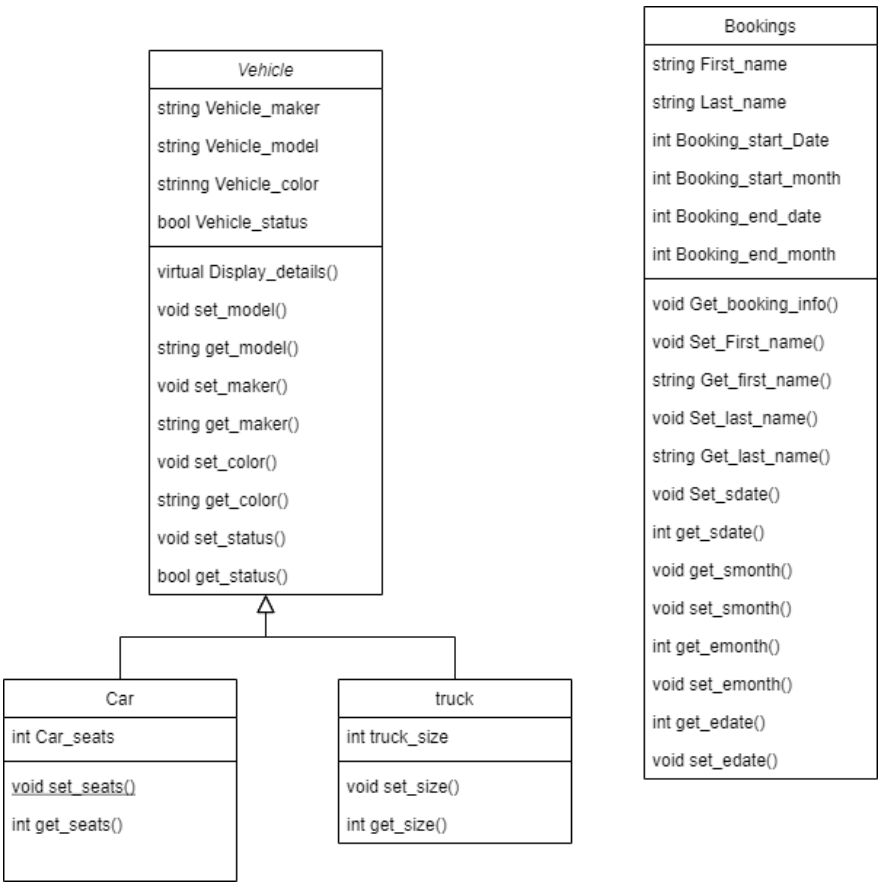
Introduction

Our project plans on implementing a vehicle rental system. It allows users to create a booking, then choose from a predefined list of available cars and trucks. User details and booking timings will be required to complete the booking. The user can also cancel booking from the main menu if required.

Design Description

1. Memory allocation from the **stack** and the **heap**
    - *Arrays* : arrays to store the information relating to the booking.
    - *Strings* : Most inputs like car type, car model, car make etc.
    - *Objects* : Vehicles, Cars, Trucks & Bookings
2. User Input and Output
    - *I/O of different data types* : Command Line.
3. Object-oriented programming and design
    - Inheritance: Vehicles, Parent Class
    - Polymorphism:
    - Abstract Classes
4. Testing
    - Test Inputs / Expected Outputs
    - Automated Testing
    - Regression Testing

Class Diagram



Initial Class Descriptions
Main Interface: Lets user choose the option they want to
Vehicle Class: Lets user choose what type of vehicle they are after and amount of seats needed.
Car Class: Lets user pick what type of car they are after (car shape, colour, seats etc)
Pickup Truck Class: Lets user pick what type of pickup truck they are after.
Booking Class: Lets user input detail to book the specific vehicle they want to book
Driver Class: Lets user input their details required to book the vehicle.

Added Class Description:
Function file: Acts as a stepping stone to access our other classes

Testing Plan

Testing will be done incrementally. Each header and class file will have an individual testing file under the name "ClassNameTest.cpp" that has the ability to test each of the functions within that class with both positive and fail outputs.

Unit Testing
Individual testing files under the default header file name, “ClassNameTest.cpp”.

Schedule Plan

Week 8
Brainstorm ideas for the final project.
Create a Discord server for the project - share code, discuss the project outside of Uni, etc.
Create a Motion project to track our progress.
Brainstorm basic class structure and files.

Week 1 Holidays
Finalise ideas via online contact (Discord).

Week 2 Holidays
Meet up in person at Uni and create visual representations of the ideas discussed in the previous week.
Finalise class diagram.
Brainstorm class and file names.
Create header files for the classes.
Document our progress on Motion.

Week 9
Finalise header files for the classes done.
Finalise a command line input system.
Create a user input system to test code.
Document our progress on Motion.

Week 10
Create a function file that acts as a stepping stool to access our header functions.

Create a main file to output the car rental system.
Continue user input testing to gain insight on former or future mistakes.
Document our progress on Motion.

Week 11
Fix any final bugs / incorrect inputs.
Finalise user input testing files to include every function.
Finalise our documentation on Motion.