# Uber

In this assignment you will implement some functionalities of Uber according to the requirements given below.

Due date:22nd April, 2019

## Design Requirements

There can be two type of Uber users, Driver and Passenger

Each user will have following attributes and behavior

Attributes

* + char\* Name
  + ? Date of Birth (see details below of data type date of birth)
  + Char \*Email
  + Char \*Phone Number
  + ? trips (an array of Trips see the detail of trip below)
  + ? Current trip (a pointer to the current trip/ride driving or riding)

Behavior

* + Change Name
  + Change DOB
  + Change phone number
  + get average rating (different for each type of user)

In addition driver will have following attributes and behavior

Attributes

* + char\* Driving License Number
  + char\* Vehicle Identification number

Behavior

* + Pick a ride (this will make the status of the trip “In progress”, only the trip that is “looking for driver” can be picked up)
  + End a ride (this will make the status of the trip “complete”)
  + Rate passenger of a trip (only for his own trips that have ended, cancelled trips cannot be rated)
  + get average rating, will return the average Drivers rating of each trip of this driver

Passenger will also have following additional information

Attributes

* + ? PaymentMethods (An array of Payment methods see details below of data type Payment Methods, max 3 payment methods)

Behaviors

* + Add/ Delete a payment method
  + Book a ride (this will create and return a trip with status “looking for driver” with passanger as this passenger and date as current date)
  + Cancel a ride (only if the ride is not complete, this will make the status of trip “cancelled”) passanger can cancel any ride if it’s not completed.
  + Rate driver of a trip (only for his own trips that have ended, cancelled rides cannot be rated)
  + get average rating, will return the average Passenger rating of each trip of this Passenger

A Trip has following attributes and behaviors

Attributes

* + Status (either looking for driver,In progress, Complete or cancelled)
  + ? Driver
  + ? Passenger
  + ? Date
  + Char \*Pick up Location
  + Char \*Drop off Location
  + Drivers rating (as given by passenger of this ride to drive of this ride between 1-5)
  + Passenger rating (as given by driver between 1-5)

Behaviors

* + Change Status
  + Rate Driver
  + Rate Passenger
  + Set pick up location
  + Set Drop of location
  + Set driver
  + Set passenger

Date will have following attributes and behavior

Attributes

* + Date
  + Month
  + Year

Behaviors

* + Get/set Date/Month/Year
  + Default constructor of Date will create date with current date (using ctime class)

Payment methods will have following attributes and behavior

Attributes

* + Type (cash or card, cannot be changed later)
  + Card Number (only if the type is car)

Behaviors

* + Get/set method for card number.

## You have to

Design you classes keeping in mind the above requirements. The following information will further help you to identify relationship of Composition, aggregation, association (bidirectional) and/or inheritance.

* The data type of most of the attributes given above as mentioned. However you will see “?” at some places. At these particular places you have to identify the datatype and whether it should be pointer or not. These relationship will be Composition, aggregation, association (bidirectional)
* Further choice between composition, aggregation, association is owning, non-owing and bidirectional non-owing relationship resp. Following hints will further help you to decide
  + Person owns it DOB, if person is destroyed you will not keep its DOB.
  + The trip will keep the information of driver and passenger and the passenger and drivers will also keep the information of their trips.
  + Person does not own its trip. For example if a driver (which is a person) is destroyed, its trip will still remain. However the information of driver in the trip should be updated.
  + A trip does not own a person, i.e. if a trip is destroyed, the driver and passenger of this trip will not be destroyed. However, this trip should be deleted from their trips.
  + Nobody should be able to create a trip except passenger. However once the passage has created a trip (or ride) he does not own it, i.e he cannot delete it.
  + Payment methods are owned by passengers, there is no point of keeping the payment info of passenger that has been destroyed.
* Identify the inheritance relationship between classes. (Hint see “is a” or “type of” to find such relations).
* Override the methods of parent class where required.
* No passenger can create a new trip if any of his/her trip is in progress.
* No driver can pick a new trip if any of his/her trip is in progress.
* You can create more functions (friend, public or utilities or get/set) where required.
* You have to insertion and extraction operator of each class. Also provide copy constructor and overload assignment operator for each class with pointer data members.
* Make sure that there is no memory leak in your code.
* Create separate header file and implementation file for each class
* You can use vector class where ever you need to keep dynamic arrays.
* Also create a class diagram as submit it with your code

Testcases.jpg file and the driver code to give you some test cases along with expected output.

Note that Uber application (i.e. driver function in this case) keeps the all the trips in trip array. There are better ways to keep this data, however right now we are just keeping a constant size array in drive code file.