***Class Relationships***

Rides

Passengers

Drivers

Collects Collects Collects

Driver

Ride

Passenger

transports takes

Economy

Basic

Group

Luxury

***Class Contents***

Driver

* Driver ID (int / 6 digits)
* Driver Name (string)
* Vehicle capacity (int)
* Handicapped Capable (Boolean)
* Vehicle Type (string)
* Rating (float / 0-5)
* Available (Boolean)
* Pets allowed (Boolean)
* Ride ID (int)
* Notes (String)
* Set/Get Driver ID
* Set/Get Name
* Set/Get Vehicle capacity
* Set/Get Handicapped
* Set/Get Vehicle type
* Set/Get Rating
* Set/Get Available
* Set/Get Pets allowed
* Set/Get Notes
* Get Rides

Passenger

* Passenger ID (int / 6 digits)
* Passenger Name (string)
* Payment Preference (string / credit, debit)
* isHandicapped (Boolean)
* Default rating (float)
* hasPets (Boolean)
* Ride ID (int)
* Set/Get Passenger ID
* Set/Get Passenger Name
* Set/Get Payment Preference
* Set/Get isHandicapped
* Set/Get Default rating
* Set/Get hasPets
* Get Rides

Ride

* Ride ID (8 digit value auto assigned)
* Pickup location (string)
* Pickup Time (Time value)
* Drop-off location (string)
* Size of party (whole number)
* Includes pets (Boolean)
* Drop-off time (Time value)
* Status (Active, Completed, Cancelled)
* Rating by customer (float)
* Set/Get Ride ID
* Set/Get Pickup Location
* Set/Get Pickup Time
* Set/Get Drop-off location
* Set/Get Drop-off Time
* Set/Get Size of Party
* Set/Get Includes Pets
* Set/Get Status
* Rating by customer

Basic

* Capacity(2-4)
* Cargo Capacity
* Set/Get Capacity
* Set/Get Cargo Capacity

Group

* Capacity(5-7)
* Cargo Capacity
* Set/Get Capacity
* Set/Get Cargo Capacity

Economy

* Capacity(< 2)
* Cargo Capacity
* Set/Get Capacity
* Set/Get Cargo Capacity

Luxury

* Capacity(7 <)
* Cargo Capacity
* Set/Get Capacity
* Set/Get Cargo Capacity
* Set/Get Has Tv
* Set/Get Has Phone
* Set/Get Has Bar
* Set/Get Has Party Light
* Set/Get Has Bluetooth
* Set/Get Has Wifi

Passengers

* Add
* Delete
* Find
* Take a ride (should call assign a ride method from rides class)
* Print all rides for passenger

Drivers

* Add
* Delete
* Find
* Print the info about his ID, Name, Vehicle capacity, Handicapped capable, Rating, Available, Pets allowed. It should be outputted in user friendly and clear format
* Print all the rides for driver

Rides

* Assign a ride id to a driver.
* Check for drivers Vehicle capacity
* Check for driver’s availability
* Check if for driver’s handicap capability
* Check for driver rating

***Function Pseudo Code***

**Drivers methods**

(these methods store all the info about registered drivers and gives an ability to a user to manipulate with the date)

Add

* Prompt user for ID(check if id used were used before)
* Prompt user for name (string)
* Prompt user vehicle capacity (int)
* Prompt user for handicapped capability (Boolean)
* Prompt user for vehicle type (string)
* Prompt user for pets allowed (Boolean)

Delete

* Prompt user for Driver ID (int 6 digits)
* Find the matching index to driver id
* Delete all the info that is on that index (name, vehicle capacity and etc.)

Find

* Prompt user for an option whether he want to look by name or id (char. n for name and I for id)
* If option is ID then:
  + Prompt user for Driver’s ID (int)
  + Loop through list of Driver’s id (use for loop and index int i)
  + If matching driver’s ID is found print to the console (it should be all the info that is matched to driver’s id like name, vehicle capacity, and etc)
* If option is Name then:
  + Prompt user for Driver’s name (string)
  + Loop through list of Driver’s names (use for loop and index int i)
  + If matching driver’s name is found print to the console (it should be all the info that is matched to driver’s id like name, vehicle capacity, and etc)

Print Info about all the drivers

* Loop through all the drivers and print them all to the console ( name, vehicle type, id, pets allowed, rating, and etc)

Print Info about rides the driver took

* Prompt the user for Driver’s ID (int)
* Loop through rides array and print to the console ( ride id, pick up location, pick up time, drop off location, drop off time)

**Passengers methods**

(these methods store all the info about registered drivers and gives an ability to a user to manipulate with the date)

Add

* Prompt user for ID(check if id used were used before)
* Prompt user for name (string)
* Prompt user for isHandicapped (Boolean)
* Prompt user for hasPets (Boolean)
* Prompt user for payment preference (string)

Delete

* Prompt user for Passenger ID (int 6 digits)
* Find the matching index to passenger id
* Delete all the info that is on that index (name, id, hasPets and etc.)

Find

* Prompt user for an option whether he want to look by name or id (char type - n for name and i for id)
* If option is ID then:
  + Prompt user for Passengers’s ID (int)
  + Loop through list of Passenger’s id (use for loop and index int i)
  + If matching passenger’s ID is found print to the console (it should be all the info that is matched to passengers’s id like name, vehicle capacity, and etc)
* If option is Name then:
  + Prompt user for Passengers’s name (string)
  + Loop through list of Passengers’s names (use for loop and index int i)
  + If matching passenger’s name is found print to the console (it should be all the info that is matched to passenger’s id like name, vehicle capacity, and etc)

Take a ride

* Call the method from Rides (get assigned a ride id)
* Print all the info about the ride (pick up time, drop off time, pick up location, drop off location)

Print Info about all the registered passengers

* Loop through all the passengers and print them all to the console ( name, vehicle type, id, pets allowed, rating, and etc)

Print Info about rides the passenger took

* Prompt the user for Passenger’s ID (int)
* Loop through rides array and print to the console ( ride id, pick up location, pick up time, drop off location, drop off time)

**Rides methods**

(these methods store all the info about the rides taken by a passenger and a driver. However, most important function of this method is to assign a ride id to a passenger and a driver)

Assign a Ride Id

* Generate a random integer number
* Assign the ride id to that random number
* Loop through all the drivers
* Check for driver’s availability (whether he has a ride it with no drop of time) returns true or false
* Check for driver’s handicap capability (whether they can carry a isHandicap passenger) returns true of false
* Check for driver rating (whether passenger is good with rating of a driver) returns the rating
* Assign the ride id to a driver.