**Description Of Classes**

1. **City :-**

This class contains the following data members :-

Rows - The number of rows in the imaginary 2D grid. (m)

Columns - The number of columns in the grid. (n).

Grid - An m\*n matrix.

It also contains the following methods :-

City() - a default constructor. This constructor does not specify the number of rows or columns of the city and hence if an object is created using this, the instance variables of that object will contain default values.

City(int rows, int columns) - This is a parameterized constructor which initializes the instance variables of the object. This constructor also initializes the grid.

These setter methods ensure that the data remains encapsulated.

setRows() - helps us set the number of rows in the city.

setColumns - helps us set the columns of the city.

makeGrid() - this method initializes the grid matrix of the object by allotting it memory appropriate to the number of rows and columns that have been taken as input.

checkLandmark(List<Landmark> l) :- This method makes sure that the given input is correct. If there exists a landmark which does not lie on our 2D grid, this method will return false and the program will terminate.

1. **Landmark :-**

This class has the following data members:

Name - The name of a landmark in the city.

Abscissa - The x coordinate of the landmark.

Ordinate - The y coordinate of the landmark.

This class also has the following methods :-

Landmark() - This is the default constructor. Whenever an object is created using this constructor, the origin is set as a landmark.

Landmark(Landmark obj) - This is the constructor which can be used to create a new Landmark object using a reference from a previously existing landmark.

setName(String s) - This method is used to set the name of the landmark.

setAbscissa(int x) - This method is used to set the x coordinate of the landmark.

setOrdinate(int y) - This method is used to set the y coordinate of the landmark.

getAbscissa() - This method returns the x coordinate of the current landmark.

getOrdinate() - This method return the y coordinate of the current landmark.

1. **Driver :-**

This class contains the following data members:

name: The name of the driver (taken as input).

rating: Performance rating of the driver (taken as input).

currentLocation: This is the landmark where driver is currently at (name of the landmark is taken as input).

available: decides the availability status of the driver. (taken as input)

speed: The speed at which driver drives the cab. (taken as input).

eta: The estimated time in which driver will pick up the customer.

notify: This helps the driver know if there are any available customers.

This class contains the following methods:

calculateEta(int pickupX, int pickupY) : This method takes the pickup location of the customer as an input and returns the ETA using the current location of our driver.

startRide() : Driver calls this method after the ride has started.

endRide() : Driver calls this method when the ride has ended.

getName() : returns the name of the driver.

setRating(float rating) : allows us to set the performance rating of the driver.

setSpeed(int speed) : allows us to set the speed of the driver.

setCurrentLocation (Landmark l) : initializes the current location of the driver.

getRating() : returns the performance rating of the driver.

setName() : allows us to set the name of the newly registered driver.

changeAvailableStatus() : Driver can call this method whenever he wants to change his availability status.

Driver() : default constructor for the driver class.

Driver(String name, float rating, Landmark currentLocation, boolean available, int speed) : Parameterized constructor of the driver class which initializes the instance variables of the driver object.

1. **Customer:**

This class contains the following data members:

name: The name of the customer.

pickupLocation : The pickup location of the customer.

destination: The destination of the customer.

This class contains the following methods:

cancelBooking(Driver d) :- When the driver has accepted the ride, this method gives a choice to the customer and allows it to cancel the booking. The program will terminate if the user decides to do so.

setDestination(String name, int x, int y) :- this method sets the destination location of the customer by creating a new Landmark object using the input parameters.

setPickupLocation(String name, int x, int y) :- this method works in a similar way to the setDestination method and initializes the pickup Location of the customer by creating a new Landmark object.