SE 471 Software Architecture

Lab 2

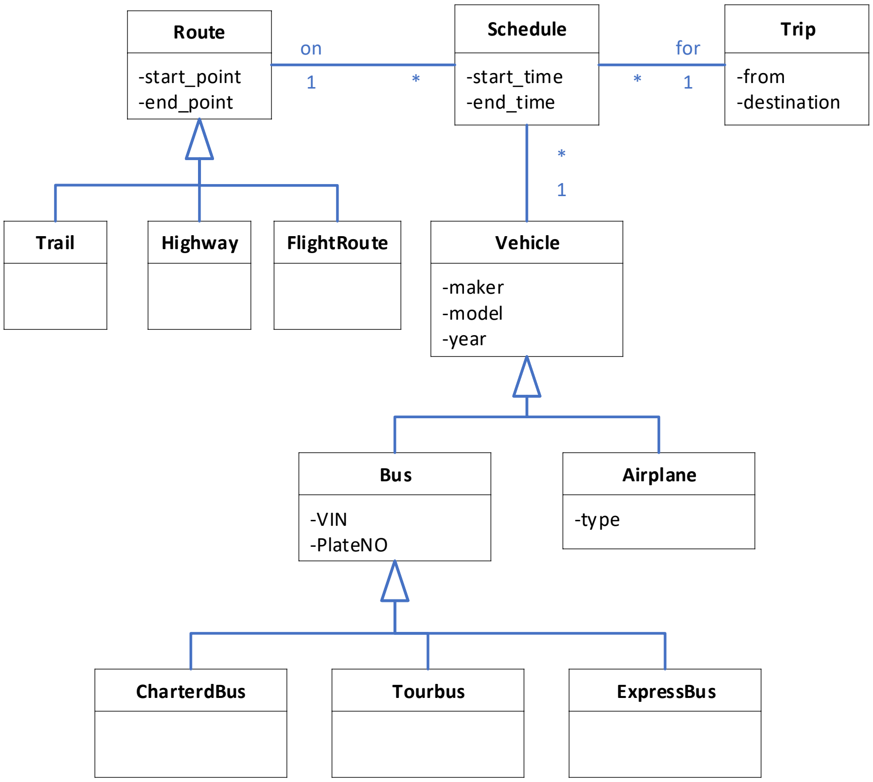
	Student Name	Student CSUSM ID	Contribution percentage
1	Lauren Gonzalez	gonza823	50
2	Sirena Murphree	murph135	50

Grading Rubrics (for instructor only):

Criteria	1. Beginning	2. Developing	3. Proficient	4. Exemplary
Mapping from design to Java code	0-9	10-14	15-19	30
Program: quality -> ***Readability***	0-2	3-5	6-9	10
Program: quality -> ***Modularity***	0-2	3-5	6-9	10
Program: quality -> ***Simplicity***	0-2	3-5	6-9	10
Updated design: ***correctness***	0-9	10-14	15-19	20
Updated design: ***Consistency with code***	0-9	10-14	15-19	20
Total Grade (100)				

Problems:

a. In the following design in UML class diagram, some classes are incomplete (lack of attributes and/or operations). You should update the class diagram by adding important attributes and/operations that are appropriate.



b. Translate your complete design into Java implementation. Remember, the goal is to make sure the implementation is consistent with the design.

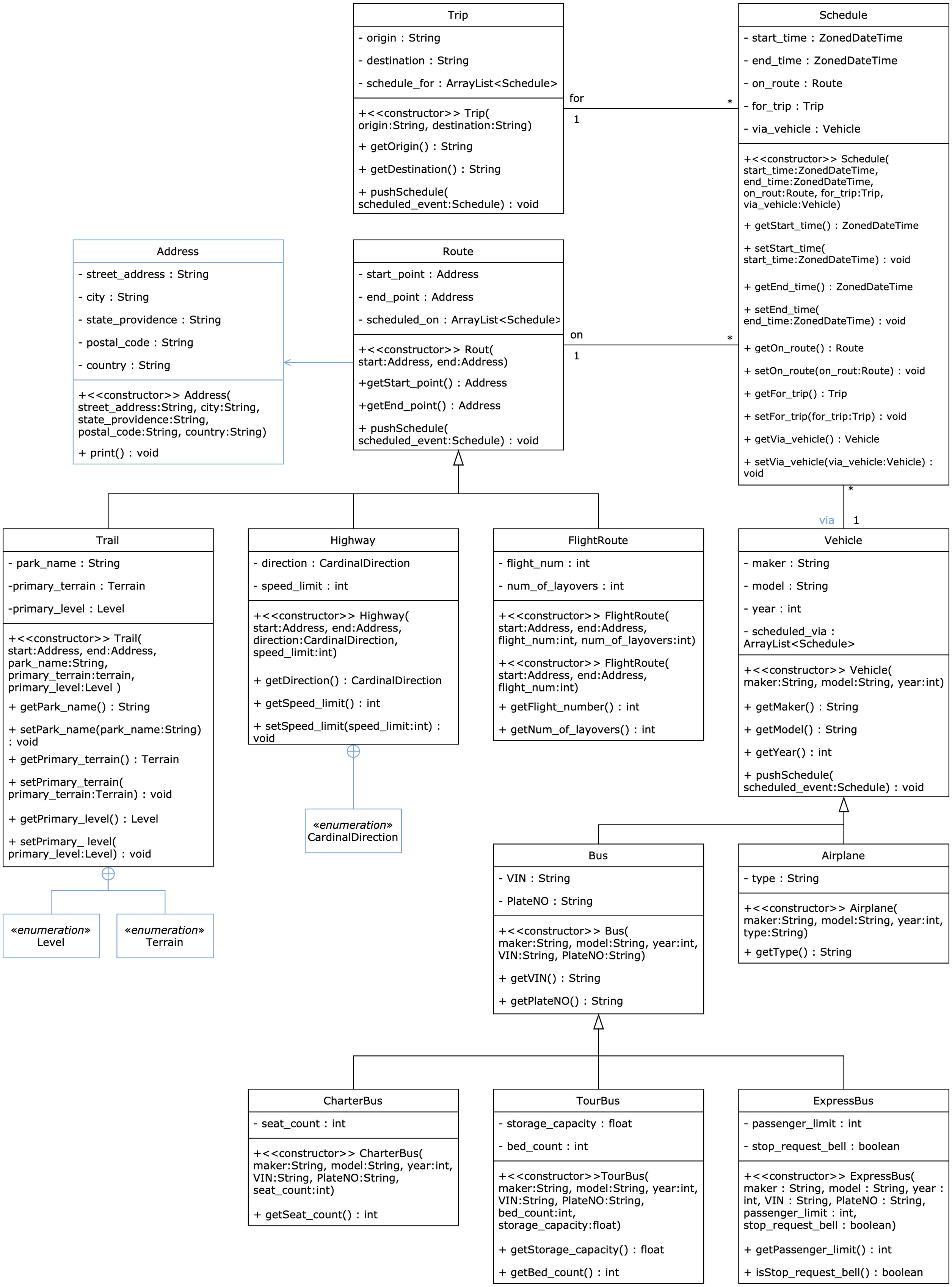
Solution:

- First, remember to zip the src folder of your project and submit the zip file to the ungraded assignment named "Lab1CodeSubmission". One submission from each team.

- Paste all you source code here.

- Paste your updated UML class diagram below.

- Save this report in PDF, then **each student** needs to submit the pdf report to the graded assignment named “Lab1ReportSubmission”.



Trip
- origin : String
- destination : String
- schedule_for : ArrayList<Schedule>
+<<constructor>> Trip(
origin:String, destination:String)
+ getOrigin() : String
+ getDestination() : String
+ pushSchedule(
scheduled_event:Schedule) : void

Schedule
- start_time : ZonedDateTime
- end_time : ZonedDateTime
- on_route : Route
- for_trip : Trip
- via_vehicle : Vehicle
+<<constructor>> Schedule(
start_time:ZonedDateTime,
end_time:ZonedDateTime,
on_rout:Route, for_trip:Trip,
via_vehicle:Vehicle)
+ getStart_time() : ZonedDateTime
+ setStart_time(
start_time:ZonedDateTime) : void
+ getEnd_time() : ZonedDateTime
+ setEnd_time(
end_time:ZonedDateTime) : void
+ getOn_route() : Route
+ setOn_route(on_rout:Route) : void
+ getFor_trip() : Trip
+ setFor_trip(for_trip:Trip) : void
+ getVia_vehicle() : Vehicle
+ setVia_vehicle(via_vehicle:Vehicle) :
void

Address
- street_address : String
- city : String
- state_providence : String
- postal_code : String
- country : String
+<<constructor>> Address(
street_address:String, city:String,
state_providence:String,
postal_code:String, country:String)
+ print() : void

Route
- start_point : Address
- end_point : Address
- scheduled_on : ArrayList<Schedule>
+<<constructor>> Rout(
start:Address, end:Address)
+getStart_point() : Address
+getEnd_point() : Address
+ pushSchedule(
scheduled_event:Schedule) : void

Trail
- park_name : String
-primary_terrain : Terrain
-primary_level : Level
+<<constructor>> Trail(
start:Address, end:Address,
park_name:String,
primary_terrain:terrain,
primary_level:Level)
+ getPark_name() : String
+ setPark_name(park_name:String)
: void
+ getPrimary_terrain() : Terrain
+ setPrimary_terrain(
primary_terrain:Terrain) : void
+ getPrimary_level() : Level
+ setPrimary_ level(
primary_level:Level) : void

Highway
- direction : CardinalDirection
- speed_limit : int
+<<constructor>> Highway(
start:Address, end:Address,
direction:CardinalDirection,
speed_limit:int)
+ getDirection() : CardinalDirection
+ getSpeed_limit() : int
+ setSpeed_limit(speed_limit:int) :
void

FlightRoute
- flight_num : int
- num_of_layovers : int
+<<constructor>> FlightRoute(
start:Address, end:Address,
flight_num:int, num_of_layovers:int)
+<<constructor>> FlightRoute(
start:Address, end:Address,
flight_num:int)
+ getFlight_number() : int
+ getNum_of_layovers() : int

Vehicle
- maker : String
- model : String
- year : int
- scheduled_via :
ArrayList<Schedule>
+<<constructor>> Vehicle(
maker:String, model:String, year:int)
+ getMaker() : String
+ getModel() : String
+ getYear() : int
+ pushSchedule(
scheduled_event:Schedule) : void

«enumeration»
CardinalDirection

«enumeration»
Level

«enumeration»
Terrain

Bus
- VIN : String
- PlateNO : String
+<<constructor>> Bus(
maker:String, model:String, year:int,
VIN:String, PlateNO:String)
+ getVIN() : String
+ getPlateNO() : String

Airplane
- type : String
+<<constructor>> Airplane(
maker:String, model:String, year:int,
type:String)
+ getType() : String

CharterBus
- seat_count : int
+<<constructor>> CharterBus(
maker:String, model:String, year:int,
VIN:String, PlateNO:String,
seat_count:int)
+ getSeat_count() : int

TourBus
- storage_capacity : float
- bed_count : int
+<<constructor>>TourBus(
maker:String, model:String, year:int,
VIN:String, PlateNO:String,
bed_count:int,
storage_capacity:float)
+ getStorage_capacity() : float
+ getBed_count() : int

ExpressBus
- passenger_limit : int
- stop_request_bell : boolean
+<<constructor>> ExpressBus(
maker : String, model : String, year :
int, VIN : String, PlateNO : String,
passenger_limit : int,
stop_request_bell : boolean)
+ getPassenger_limit() : int
+ isStop_request_bell() : boolean

Schedule.java

```
import java.time.ZonedDateTime;

public class Schedule {

      /**
       * departure time
       */
      private ZonedDateTime start_time;

      /**
       * arrival time
       */
      private ZonedDateTime end_time;

      /**
       * rout taken
       */
      private Route on_route;

      /**
       * trip this schedule is for
       */
      private Trip for_trip;

      /**
       * vehicle used
       */
      private Vehicle via_vehicle;

      /**
       * constructor
       * @param start_time    departure time
       * @param end_time      arrival time
       * @param on_route      rout taken
       * @param for_trip      trip this schedule is for
       * @param via_vehicle   vehicle used
       */
      public Schedule(ZonedDateTime start_time, ZonedDateTime end_time, Route
on_route, Trip for_trip, Vehicle via_vehicle) {
            this.setStart_time(start_time);
            this.setEnd_time(end_time);
            this.setOn_route(on_route);
            this.setFor_trip(for_trip);
            this.setVia_vehicle(via_vehicle);
      }

      /**
       * @return the start_time
```

```
     */
    public ZonedDateTime getStart_time() {
          return start_time;
    }

    /**
     * @param start_time the start_time to set
     */
    public void setStart_time(ZonedDateTime start_time) {
          this.start_time = start_time;
    }

    /**
     * @return the end_time
     */
    public ZonedDateTime getEnd_time() {
          return end_time;
    }

    /**
     * @param end_time the end_time to set
     */
    public void setEnd_time(ZonedDateTime end_time) {
          this.end_time = end_time;
    }

    /**
     * @return the on_route
     */
    public Route getOn_route() {
          return on_route;
    }

    /**
     * @param on_route the on_route to set
     */
    public void setOn_route(Route on_route) {
          this.on_route = on_route;
    }

    /**
     * @return the for_trip
     */
    public Trip getFor_trip() {
          return for_trip;
    }

    /**
     * @param for_trip the for_trip to set
     */
    public void setFor_trip(Trip for_trip) {
          this.for_trip = for_trip;
```

```
      }

      /**
       * @return the via_vehicle
       */
      public Vehicle getVia_vehicle() {
            return via_vehicle;
      }

      /**
       * @param via_vehicle the via_vehicle to set
       */
      public void setVia_vehicle(Vehicle via_vehicle) {
            this.via_vehicle = via_vehicle;
      }
}
```

Trip.java

```
import java.util.ArrayList;

public class Trip {

      /**
       * general origin of trip
       */
      private String origin;

      /**
       * general destination of trip
       */
      private String destination;

      /**
       * list of schedules the trip will take
       */
      private ArrayList<Schedule> scheduled_for;

      /**
       * constructor
       * @param origin  the trip's origin
       * @param destination    the trip's destination
       */
      public Trip(String origin, String destination) {
            this.origin = origin;
            this.destination = destination;
            this.scheduled_for = new ArrayList<Schedule>();
      }

      /**
       * @return the origin
```

```java
     */
    public String getOrigin() {
        return origin;
    }

    /**
     * @return the destination
     */
    public String getDestination() {
        return destination;
    }

    /**
     * @param scheduled_event the Schedule to be added to scheduled_for
     */
    public void pushSchedule(Schedule scheduled_event) {
        this.scheduled_for.add(scheduled_event);
    }
}
```

Route.java

```java
import java.util.ArrayList;

public class Route {

    /**
     * the Address of the route's start_point
     */
    private Address start_point;

    /**
     * the Address of the route's end_point
     */
    private Address end_point;

    /**
     * the list of schedules that take this route
     */
    private ArrayList<Schedule> scheduled_on;

    /**
     * constructor
     * @param start the Address of the route's start_point
     * @param end     the Address of the route's end_point
     */
    public Route(Address start, Address end) {
        this.start_point = start;
        this.end_point = end;
        this.scheduled_on = new ArrayList<Schedule>();
    }
    /**
     * @return the start_point
     */
    public Address getStart_point() {
        return start_point;
    }

    /**
     * @return the end_point
     */
    public Address getEnd_point() {
        return end_point;
    }

    /**
     * @param scheduled_event the Schedule to be added to scheduled_on
     */
    public void pushSchedule(Schedule scheduled_event) {
        this.scheduled_on.add(scheduled_event);
    }
}
```

Address.java

```
public class Address {

    /**
     * street address
     * "285 J St"
     */
    private String street_address;

    /**
     * city
     * "San Diego"
     */
    private String city;

    /**
     * state or providence
     * "California"
     */
    private String state_providence;

    /**
     * postal code
     * "92101"
     */
    private String  postal_code;
    /**
     * country
     * "USA"
     */
    private String country;

    /**
     * constructor
     * @param street_address
     * @param city
     * @param state_providence
     * @param postal_code
     * @param country
     */
    public Address(String street_address, String city, String
state_providence, String postal_code, String country) {
        this.street_address = street_address;
        this.city = city;
        this.state_providence = state_providence;
        this.postal_code = postal_code;
        this.country = country;
    }

    /**
     * print formated address
     * 285 J St
     * San Diego, CA 92101
     * USA
     */
    public void print() {
        System.out.printf("%s\n%s, %s %s\n%s", street_address,  city,
state_providence, postal_code, country);
    }
}
```

Trail.java

```
public class Trail extends Route {

    /**
     * enumeration type of different terrains a trail can be
     */
    public enum Terrain {
        FOOT_TRAIL,
        BIKEWAY,
        BOARDWALK,
        NATURE,
        MULTI_USE

    }
```

```
      /**
       * enumeration different levels of experience required for hiking a
trail
       */

      public enum Level{
            BEGINNER,
            INTERMEDIATE,
            SKILLED,
            EXPERT
      }

      /**
       * name of the park the trail goes through
       */
      private String park_name;

      /**
       * terrain of the trail
       */
      private Terrain primary_terrain;

      private Level primary_level;

      /**
       * constructor
       * @param start the Address of the route's start_point
       * @param end     the Address of the route's end_point
       * @param park_name     name of the park the trail goes through
       * @param primary_terrain     terrain of the trail
       * @param primary_level       level of the trail
       */
      public Trail(Address start, Address end, String park_name, Terrain
primary_terrain, Level primary_level) {
            super(start, end);
            this.setPark_name(park_name);
            this.setPrimary_terrain(primary_terrain);
            this.setPrimary_level(primary_level);
      }

      /**
       * @return the park_name
       */
      public String getPark_name() {
            return park_name;
      }


      /**
       * @param park_name the park_name to set
       */
      public void setPark_name(String park_name) {
            this.park_name = park_name;
      }

      /**
       * @return the primary_terrain
       */
      public Terrain getPrimary_terrain() {
            return primary_terrain;
      }

      /**
       * @param primary_terrain the primary_terrain to set
       */
      public void setPrimary_terrain(Terrain primary_terrain) {
            this.primary_terrain = primary_terrain;
      }

      /**
       * @return the primary_level
       */

      public Level getPrimary_Level() {
            return primary_level;
      }

      /**
       * @param primary_level the primary_level to set
       */

      public void setPrimary_level(Level primary_level) {
            this.primary_level = primary_level;
      }

}
```

Highway.java

```
public class Highway extends Route {

    /**
     * enumeration of the Cardinal Directions
     */
    enum CardinalDirection{
        NORTH,
        SOUTH,
        EAST,
        WEST,
        NORTHWEST,
        NORTHEAST,
        SOUTHWEST,
        SOUTHEAST
    }
```

```
      /**
       * general direction to travel down the highway
       */
      private CardinalDirection direction;

      /**
       * the speed limit of the highway
       */
      private int speed_limit;

      /**
       * constructor
       * @param start the Address of the route's start_point
       * @param end     the Address of the route's end_point
       * @param direction     general direction to travel down the highway
       * @param speed_limit   the speed limit of the highway
       */
      public Highway(Address start, Address end, CardinalDirection direction,
int speed_limit) {
            super(start, end);
            this.direction = direction;
            this.speed_limit = speed_limit;
      }

      /**
       * @return the direction
       */
      public CardinalDirection getDirection() {
            return direction;
      }

      /**
       * @return the speed_limit
       */
      public int getSpeed_limit() {
            return speed_limit;
      }
}
```

FlightRoute.java

```
public class FlightRoute extends Route {

      /**
       * flight number assigned to flight route
       */
      private int flight_num;

      /**
       * number of layovers
       * 0 if direct flight
```

```
         */
        private int num_of_layovers;

        /**
         * constructor
         * @param start the Address of the route's start_point
         * @param end     the Address of the route's end_point
         * @param flight_num    flight number assigned to flight route
         * @param num_of_layovers     number of layovers
         */
        public FlightRoute(Address start, Address end, int flight_num, int
num_of_layovers) {
                super(start, end);
                this.flight_num = flight_num;
                this.num_of_layovers = num_of_layovers;
        }

        /**
         * constructor
         * @param start the Address of the route's start_point
         * @param end     the Address of the route's end_point
         * @param flight_num    flight number assigned to route
         */
        public FlightRoute(Address start, Address end, int flight_num) {
                this(start, end, flight_num, 0);
        }

        /**
         * @return the flight_number
         */
        public int getFlight_number() {
                return flight_num;
        }

        /**
         * @return the num_of_layovers
         */
        public int getNum_of_layovers() {
                return num_of_layovers;
        }
}
```

Vehicle.java

```
import java.util.ArrayList;

public class Vehicle {

        /**
         * vehicle maker
         */
```

```java
    private String maker;

    /**
     * vehicle model
     */
    private String model;

    /**
     * vehicle manufactured year
     */
    private int year;

    /**
     * list of schedules that vehicle is scheduled for
     */
    private  ArrayList<Schedule> scheduled_via;

    /**
     * constructor
     * @param maker vehicle maker
     * @param model vehicle model
     * @param year    vehicle manufactured year
     */
    public Vehicle(String maker, String model, int year) {
          this.maker = maker;
          this.model = model;
          this.year = year;
          this.scheduled_via = new ArrayList<Schedule>();
    }

    /**
     * @return the maker
     */
    public String getMaker() {
          return maker;
    }

    /**
     * @return the model
     */
    public String getModel() {
          return model;
    }

    /**
     * @return the year
     */
    public int getYear() {
          return year;
    }

    /**
```

```
       * @param scheduled_event the Schedule to be added to scheduled_via
       */
      public void pushSchedule(Schedule scheduled_event) {
            this.scheduled_via.add(scheduled_event);
      }
}
```

Airplane.java

```
public class Airplane extends Vehicle{

      /**
       * type of airplane
       */
      private String type;

      /**
       * constructor
       * @param maker vehicle maker
       * @param model vehicle model
       * @param year    vehicle model year
       * @param type    type of airplane
       */
      public Airplane(String maker, String model, int year, String type) {
            super(maker, model, year);
            this.type = type;
      }

      /**
       * @return the type
       */
      public String getType() {
            return type;
      }
}
```

Bus.java

```
public class Bus extends Vehicle{

      /**
       * vehicle identification number
       */
      private String VIN;

      /**
       * registered license plate number
       */
      private String PlateNO;
      /**
       * constructor
       * @param maker vehicle maker
       * @param model vehicle model
       * @param year    vehicle model year
       * @param VIN     vehicle identification number
       * @param PlateNO registered license plate number
       */
      public Bus(String maker, String model, int year, String VIN, String 
PlateNO) {
            super(maker, model, year);
            this.PlateNO = PlateNO;
            this.VIN = VIN;
      }

      /**
       * @return the VIN
       */
      public String getVIN() {
            return VIN;
      }

      /**
       * @return the plateNO
       */
      public String getPlateNO() {
            return PlateNO;
      }
}
```

CharterBus.java

```
public class CharterBus extends Bus {

      /**
       * number of passenger seats
       */
      private int seat_count;

      /**
       * constructor
       * @param maker vehicle maker
       * @param model vehicle model
       * @param year    vehicle model year
       * @param VIN     vehicle identification number
       * @param PlateNO registered license plate number
       * @param seat_count    number of passenger seats
       */
      public CharterBus(String maker, String model, int year, String VIN, 
String PlateNO, int seat_count) {
            super(maker, model, year, VIN, PlateNO);
```

```
        this.seat_count = seat_count;

    }

    /**
     * @return the seat_count
     */
    public int getSeat_count() {
        return seat_count;
    }
}
```

TourBus.java

```
public class TourBus extends Bus {

    /**
     * number of beds
     */
    private int bed_count;

    /**
     * under-cab storage capacity
     */
    private float storage_capacity;

    /**
     * constructor
     * @param maker vehicle maker
     * @param model vehicle model
     * @param year    vehicle model year
     * @param VIN     vehicle identification number
     * @param PlateNO registered license plate number
     * @param bed_count     number of beds
     * @param storage_capacity    under-cab storage capacity
     */
    public TourBus(String maker, String model, int year, String VIN, String
PlateNO, int bed_count, float storage_capacity) {
        super(maker, model, year, VIN, PlateNO);
        this.bed_count = bed_count;
        this.storage_capacity = storage_capacity;
    }

    /**
     * @return the storage_capacity
     */
    public float getStorage_capacity() {
        return storage_capacity;
    }

    /**
```

```
       * @return the bed_count
       */
      public int getBed_count() {
            return bed_count;
      }
}
```

ExpressBus.java

```
public class ExpressBus extends Bus {

      /**
       * safe passenger occupancy limit
       */
      private int passenger_limit;

      /**
       * Mechanism exist that facilitates passenger to request a stop
       */
      private boolean stop_request_bell;

      /**
       * constructor
       * @param maker vehicle maker
       * @param model vehicle model
       * @param year    vehicle model year
       * @param VIN     vehicle identification number
       * @param PlateNO registered license plate number
       * @param passenger_limit     safe passenger occupancy limit
       */
      public ExpressBus(String maker, String model, int year, String VIN,
String PlateNO, int passenger_limit, boolean stop_request_bell) {
            super(maker, model, year, VIN, PlateNO);
            this.passenger_limit = passenger_limit;
            this.stop_request_bell = stop_request_bell;
      }

      /**
       * @return the passenger_limit
       */
      public int getPassenger_limit() {
            return passenger_limit;
      }

      /**
       * @return the stop_request_bell
       */
      public boolean isStop_request_bell() {
            return stop_request_bell;
      }
}
```