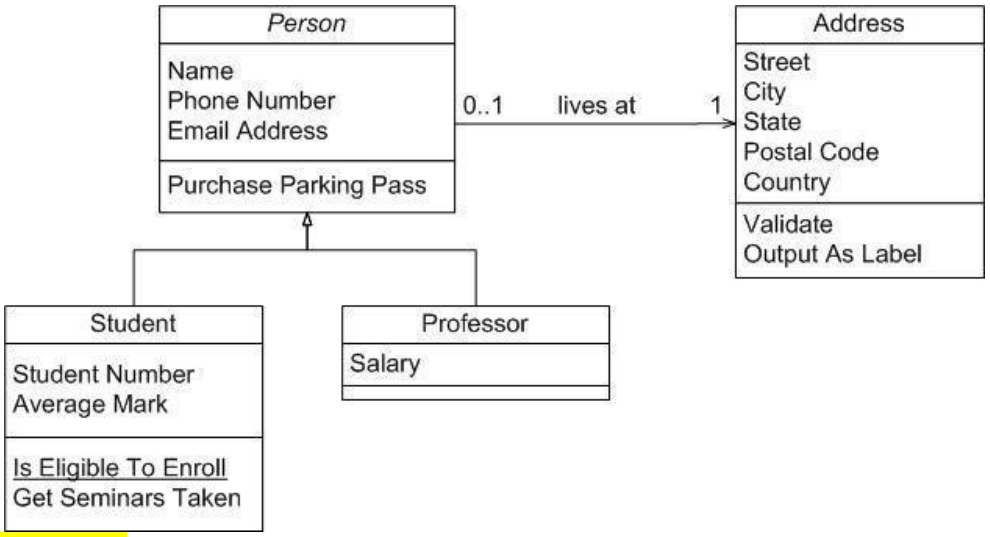
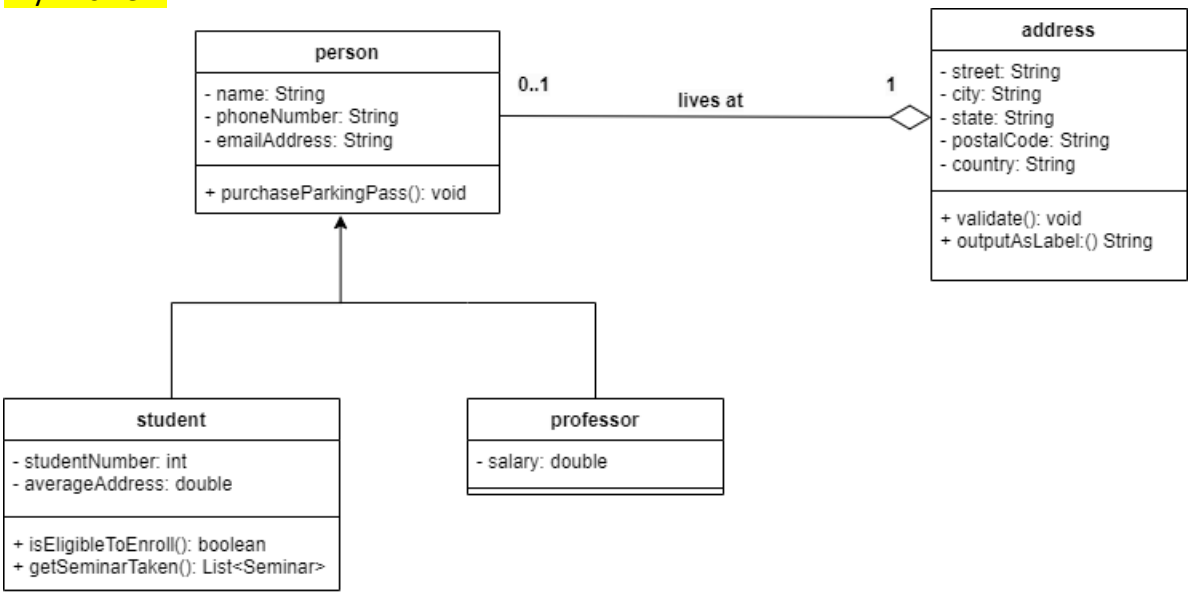
Name : Evan Diantha Fafian
Class : SIB 2G
Absent : 09
NIM : 2341760163

UTS QUESTIONS
OBJECT-BASED PROGRAMMING PRACTICUM

1. Identify the following diagram class, make complete improvements and in accordance with the rules for writing the diagram class.

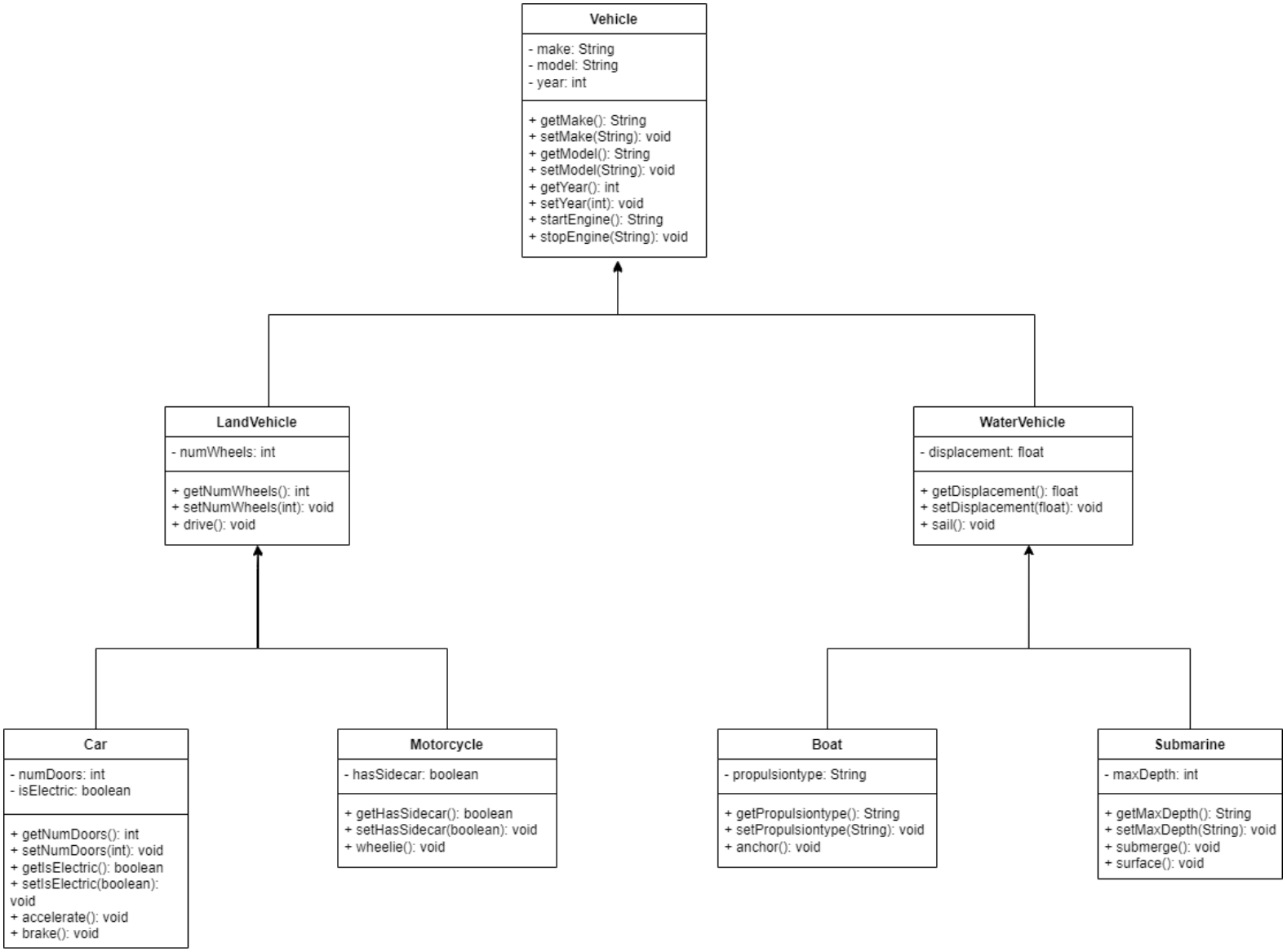


My Answer :



2. Create a diagram class that uses multilevel inheritance and create the program code!

My Answer :


Vehicle
- make: String
- model: String
- year: int
+ getMake(): String
+ setMake(String): void
+ getModel(): String
+ setModel(String): void
+ getYear(): int
+ setYear(int): void
+ startEngine(): String
+ stopEngine(String): void
LandVehicle
- numWheels: int
+ getNumWheels(): int
+ setNumWheels(int): void
+ drive(): void
WaterVehicle
- displacement: float
+ getDisplacement(): float
+ setDisplacement(float): void
+ sail(): void
Car
- numDoors: int
- isElectric: boolean
+ getNumDoors(): int
+ setNumDoors(int): void
+ getIsElectric(): boolean
+ setIsElectric(boolean): void
+ accelerate(): void
+ brake(): void
Motorcycle
- hasSidecar: boolean
+ getHasSidecar(): boolean
+ setHasSidecar(boolean): void
+ wheelie(): void
Boat
- propulsiontype: String
+ getPropulsiontype(): String
+ setPropulsiontype(String): void
+ anchor(): void
Submarine
- maxDepth: int
+ getMaxDepth(): String
+ setMaxDepth(String): void
+ submerge(): void
+ surface(): void

- Vehicle

```java
package week_8_UTS.question_2;

public class Vehicle {
    private String make;
    private String model;
    private int year;

    public Vehicle(String make, String model, int year) {
        this.make = make;
        this.model = model;
        this.year = year;
    }

    public String getMake() {
        return make;
    }

    public void setMake(String make) {
        this.make = make;
    }

    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }

    public void startEngine() {
        System.out.println("Starting the engine of " + make + " " + model);
    }

    public void stopEngine() {
        System.out.println("Stopping the engine of " + make + " " + model);
    }
}
```

- LandVehicle

```java
package week_8_UTS.question_2;

public class LandVehicle extends Vehicle {
    private int numWheels;

    public LandVehicle(String make, String model, int year, int numWheels) {
        super(make, model, year);
        this.numWheels = numWheels;
    }

    public int getNumWheels() {
        return numWheels;
    }

    public void setNumWheels(int numWheels) {
        this.numWheels = numWheels;
    }

    public void drive() {
        System.out.println("Driving the " + getMake() + " " + getModel() + " on " + numWheels + " wheels");
    }
}

```

- WaterVehicle

```java
package week_8_UTS.question_2;

public class WaterVehicle extends Vehicle {
    private float displacement;

    public WaterVehicle(String make, String model, int year, float displacement) {
        super(make, model, year);
        this.displacement = displacement;
    }

    public float getDdisplacement() {
        return displacement;
    }

    public void setDisplacement(float displacement) {
        this.displacement = displacement;
    }

    public void sail() {
        System.out.println("Sailing the " + getMake() + " " + getModel() + " with " + displacement + " displacement");
    }
}
```

- Car

```java
package week_8_UTS.question_2;

public class Car extends LandVehicle {
    private int numDoors;
    private boolean isElectric;

    public Car(String make, String model, int year, int numWheels, int numDoors, boolean isElectric) {
        super(make, model, year, numWheels);
        this.numDoors = numDoors;
        this.isElectric = isElectric;
    }

    public int getNumDoors() {
        return numDoors;
    }

    public void setNumDoors(int numDoors) {
        this.numDoors = numDoors;
    }

    public boolean isElectric() {
        return isElectric;
    }

    public void setElectric(boolean electric) {
        isElectric = electric;
    }

    public void accelerate() {
        System.out.println("Accelerating the " + getMake() + " " + getModel());
    }

    public void brake() {
        System.out.println("Braking the " + getMake() + " " + getModel());
    }
}
```

- Motorcycle

```java
package week_8_UTS.question_2;

public class Motorcycle extends LandVehicle {
    private boolean hasSidecar;

    public Motorcycle(String make, String model, int year, int numWheels, boolean hasSidecar) {
        super(make, model, year, numWheels);
        this.hasSidecar = hasSidecar;
    }

    public boolean hasSidecar() {
        return hasSidecar;
    }

    public void setHasSidecar(boolean hasSidecar) {
        this.hasSidecar = hasSidecar;
    }

    public void wheelie() {
        if (!hasSidecar) {
            System.out.println("Performing a wheelie on the " + getMake() + " " + getModel());
        } else {
            System.out.println("Cannot perform a wheelie with a sidecar on the " + getMake() + " " + getModel());
        }
    }
}
```

- Boat

```java
package week_8_UTS.question_2;

public class Boat extends WaterVehicle {
    private String propulsionType;

    public Boat(String make, String model, int year, float displacement, String propulsionType) {
        super(make, model, year, displacement);
        this.propulsionType = propulsionType;
    }

    public String getPropulsionType() {
        return propulsionType;
    }

    public void setPropulsionType(String propulsionType) {
        this.propulsionType = propulsionType;
    }

    public void anchor() {
        System.out.println("Anchoring the " + getMake() + " " + getModel());
    }
}
```

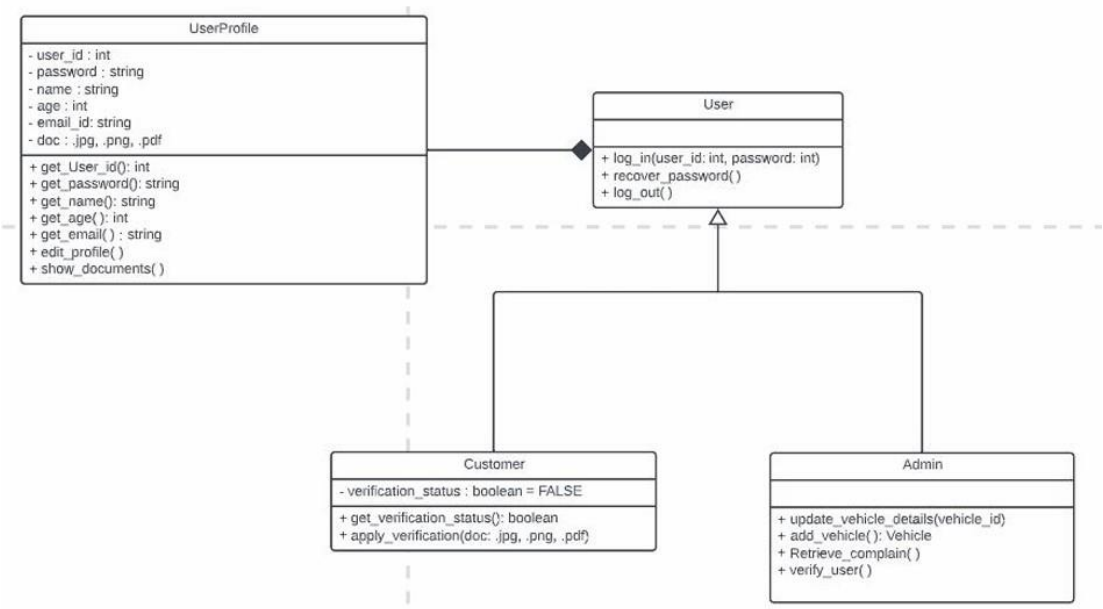
- Submarine

```java
package week_8_UTS.question_2;

public class Submarine extends WaterVehicle {
    private int maxDepth;

    public Submarine(String make, String model, int year, float displacement, int maxDepth) {
        super(make, model, year, displacement);
        this.maxDepth = maxDepth;
    }

    public int getMaxDepth() {
        return maxDepth;
    }

    public void setMaxDepth(int maxDepth) {
        this.maxDepth = maxDepth;
    }

    public void submerge() {
        System.out.println(
                "Submerging the " + getMake() + " " + getModel() + " to a maximum depth of " + maxDepth + " meters");
    }

    public void surface() {
        System.out.println("Surfacing the " + getMake() + " " + getModel());
    }
}
```

- Main

```java
package week_8_UTS.question_2;

public class MainDemoVehicle {
    public static void main(String[] args) {
        // Demonstrate Car
        System.out.println("=== Car Demo ===");
        Car car = new Car("Tesla", "Model S", 2023, 4, 4, true);
        demonstrateCar(car);

        // Demonstrate Motorcycle
        System.out.println("\n=== Motorcycle Demo ===");
        Motorcycle motorcycle = new Motorcycle("Harley Davidson", "Sportster", 2023, 2, false);
        demonstrateMotorcycle(motorcycle);

        // Demonstrate Boat
        System.out.println("\n=== Boat Demo ===");
        Boat boat = new Boat("Sea Ray", "Sundancer", 2023, 10000f, "Inboard");
        demonstrateBoat(boat);

        // Demonstrate Submarine
        System.out.println("\n=== Submarine Demo ===");
        Submarine submarine = new Submarine("General Dynamics", "Virginia class", 2023, 7800f, 800);
        demonstrateSubmarine(submarine);
    }

    private static void demonstrateCar(Car car) {
        System.out.println("Car: " + car.getMake() + " " + car.getModel() + " (" + car.getYear() + ")");
        System.out.println("Number of doors: " + car.getNumDoors());
        System.out.println("Is electric: " + car.isElectric());
        car.startEngine();
        car.drive();
        car.accelerate();
        car.brake();
        car.stopEngine();
    }

    private static void demonstrateMotorcycle(Motorcycle motorcycle) {
        System.out.println("Motorcycle: " + motorcycle.getMake() + " " + motorcycle.getModel() + " ("
                + motorcycle.getYear() + ")");
        System.out.println("Has sidecar: " + motorcycle.hasSidecar());
        motorcycle.startEngine();
        motorcycle.drive();
        motorcycle.wheelie();
        motorcycle.stopEngine();
    }

    private static void demonstrateBoat(Boat boat) {
        System.out.println("Boat: " + boat.getMake() + " " + boat.getModel() + " (" + boat.getYear() + ")");
        System.out.println("Displacement: " + boat.getDisplacement() + " tons");
        System.out.println("Propulsion type: " + boat.getPropulsionType());
        boat.startEngine();
        boat.sail();
        boat.anchor();
        boat.stopEngine();
    }

    private static void demonstrateSubmarine(Submarine submarine) {
        System.out.println(
                "Submarine: " + submarine.getMake() + " " + submarine.getModel() + " (" + submarine.getYear() + ")");
        System.out.println("Displacement: " + submarine.getDisplacement() + " tons");
        System.out.println("Maximum depth: " + submarine.getMaxDepth() + " meters");
        submarine.startEngine();
        submarine.sail();
        submarine.submerge();
        submarine.surface();
        submarine.stopEngine();
    }
}
```

3. Please identify the class diagram by providing an explanation of the concept of inheritance, the relationship between classes and the following system flow, create a program code from the following class diagram!



My Answer :

- UserProfile

```java
package week_8_UTS.question_3;

public class UserProfile09 {
    private int userId;
    private String password;
    private String name;
    private int age;
    private String emailId;
    private String doc;

    public UserProfile09(int userId, String password, String name, int age, String emailId, String doc) {
        this.userId = userId;
        this.password = password;
        this.name = name;
        this.age = age;
        this.emailId = emailId;
        this.doc = doc;
    }

    public int getUserId() {
        return userId;
    }

    public String getPassword() {
        return password;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getEmail() {
        return emailId;
    }

    public void editProfile(String newName, int newAge, String newEmail) {
        this.name = newName;
        this.age = newAge;
        this.emailId = newEmail;
        System.out.println("Profile updated successfully.");
    }

    public void showDocuments() {
        System.out.println("Documents: " + doc);
    }

    public String getDoc() {
        return doc;
    }
}
```

- User

```java
package week_8_UTS.question_3;

public class User09 {
    public boolean logIn(int userId, String password) {
        System.out.println("User  logged in with ID: " + userId);
        return true;
    }

    public void recoverPassword() {
        System.out.println("Password recovery initiated.");
    }

    public void logOut() {
        System.out.println("User  logged out.");
    }
}
```

- Customer

```java
package week_8_UTS.question_3;

public class Customer09 extends User09 {
    private boolean verificationStatus = false;

    public boolean getVerificationStatus() {
        return verificationStatus;
    }

    public void applyVerification(String doc) {
        if (doc != null && !doc.isEmpty()) {
            verificationStatus = true;
            System.out.println("Verification applied with document: " + doc);
        } else {
            System.out.println("Document not valid for verification.");
        }
    }
}
```

- Admin

```java
package week_8_UTS.question_3;

public class Admin09 extends User09 {
    public void updateVehicleDetails(int vehicleId) {
        System.out.println("Vehicle details updated for Vehicle ID: " + vehicleId);
    }

    public void addVehicle() {
        System.out.println("New vehicle added.");
    }

    public void retrieveComplain() {
        System.out.println("Retrieving user complaints.");
    }

    public void verifyUser(Customer09 customer) {
        if (customer.getVerificationStatus()) {
            System.out.println("Customer verified.");
        } else {
            System.out.println("Customer not verified.");
        }
    }
}
```

- Main

```java
package week_8_UTS.question_3;

public class Main09 {
    public static void main(String[] args) {
        UserProfile09 userProfile = new UserProfile09(101, "password101", "Emil", 30, "emil@gmail.com", "profile.jpg");
        System.out.println("User  Profile:");
        System.out.println("-".repeat(30));
        System.out.println("Name: " + userProfile.getName());
        System.out.println("Age: " + userProfile.getAge());
        System.out.println("Email: " + userProfile.getEmail());
        System.out.println("Documents: " + userProfile.getDoc());
        userProfile.showDocuments();
        userProfile.editProfile("Edo", 31, "edo@gmail.com");

        Customer09 customer = new Customer09();
        System.out.println("\nLogin sebagai Customer:");
        System.out.println("-".repeat(30));
        customer.logIn(101, "password101");
        customer.applyVerification("member_card.png");
        System.out.println("Verification Status: " + customer.getVerificationStatus());
        customer.logOut();

        Admin09 admin = new Admin09();
        System.out.println("\nLogin sebagai Admin:");
        System.out.println("-".repeat(30));
        admin.logIn(000, "admin");
        admin.addVehicle();
        admin.updateVehicleDetails(010);
        admin.retrieveComplain();
        admin.verifyUser(customer);
        admin.logOut();
    }
}
```

---- Good Luck ----