# MIDTERM ASSIGNMENT REPORT

Major: **INFORMATION TECHNOLOGY**

Course code: **CBOP3203**

Course name: **OBJECT ORIENTED PROGRAMING**

Lecturer's name: LE THI NGOC THO

Student's name:

1. Tạ Duy Thành Tài ID:2254030103 Class: 22BOIT02

Ho Chi Minh City, March 2024

**SOCIALIST REPUBLIC OF VIETNAM**

**Independence – Freedom – Happiness**

---

# COMMENTS OF THE LECTURER

**Student's full name:**

**(1)**Tạ Duy Thành Tài **ID:** 2254030103

**Lecturer's comments:**

*HCMC, ……. (date)*

*Please sign here*

**Le Thi Ngoc Tho**

I. Analize the problem

QUESTION 1a:

- Attributes/Data fields: model, yearMade, carNumber (all private)

- Methods:

+ A constructor to initialize model and yearMade

+ A constructor to initialize model, yearMade, and carNumber

+ Getter methods for each attribute

+ A setter method for carNumber

+ A method displayDetails() to display the car details

+ A method compareCarNumber() that compares the car number with another car

+ Additional Method Justification: In this methods, I'm choosing the age of the car to consider the car is vintage or not.

QUESTION 1b:

- Create two Car objects in a main() method with specified initial values.

- Display attribute values of each object.

- Use the compareCarNumber() method to compare the car numbers and display an appropriate message.

II. Copy of code

a. Car() class:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this
template
 */
package car;

/**
 *
 * @author Thanh Tai
 */
public class Car {
    private String model;
    private int yearMade;
    private String carNumber;

public Car(String model, int yearMade){
    this.model = model;
    this.yearMade = yearMade;
```

```
}
public Car(String model, int yearMade, String carNumber){
  this.model = model;
  this.yearMade = yearMade;
  this.carNumber = carNumber;
}
public String get_model(){
    return model;
}
public int get_yearMade(){
    return yearMade;
}
public String get_carNumber(){
    return carNumber;
}
public void set_carNumber(String carNumber){
    this.carNumber = carNumber;
}
public void displayDetails(){
    System.out.println("Model: " + model );
    System.out.println("Year Made: " + yearMade );
    System.out.println("Car Number: " + carNumber );
}
public void compare_carNumber(Car AnotherCar){
    if (this.carNumber.equals(AnotherCar.get_carNumber())){
        System.out.println("The cars have the same number " + carNumber);
    } else {
        System.out.println("The cars have different number " + carNumber);
}
}
public boolean Vintage(){
    final int vintage_age = 20;
    int nowYear = java.time.Year.now().getValue();
    return (nowYear - yearMade)>= vintage_age;
}
}
```

Justification: The Vintage() method adds value to the Car class by allowing users to easily check if their car is old enough to be considered vintage. This is helpful for car owners who want to join vintage car shows or need to know if their car fits the vintage category for other reasons, like insurance. It makes the class more useful by providing this extra piece of information without needing extra steps or calculations.

b. Main() class:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
```

```java
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this
template
 */
package car;

/**
 *
 * @author Thanh Tai
 */
public class Main{

    public static void main(String[] args){
        Car car1 = new Car ("Honda City", 2022, "VJM1775");
        Car car2 = new Car ("Toyota Vios", 2022, "VME841");

        car1.displayDetails();
        car2.displayDetails();

        car1.compare_carNumber(car2);
    }
}
```

Output photo:

This is I received when I run Main() class:

```
Model: Honda City
Year Made: 2022
Car Number: VJM1775
Model: Toyota Vios
Year Made: 2022
Car Number: VME841
The cars have different number VJM1775
```