

# Programming Fundamentals

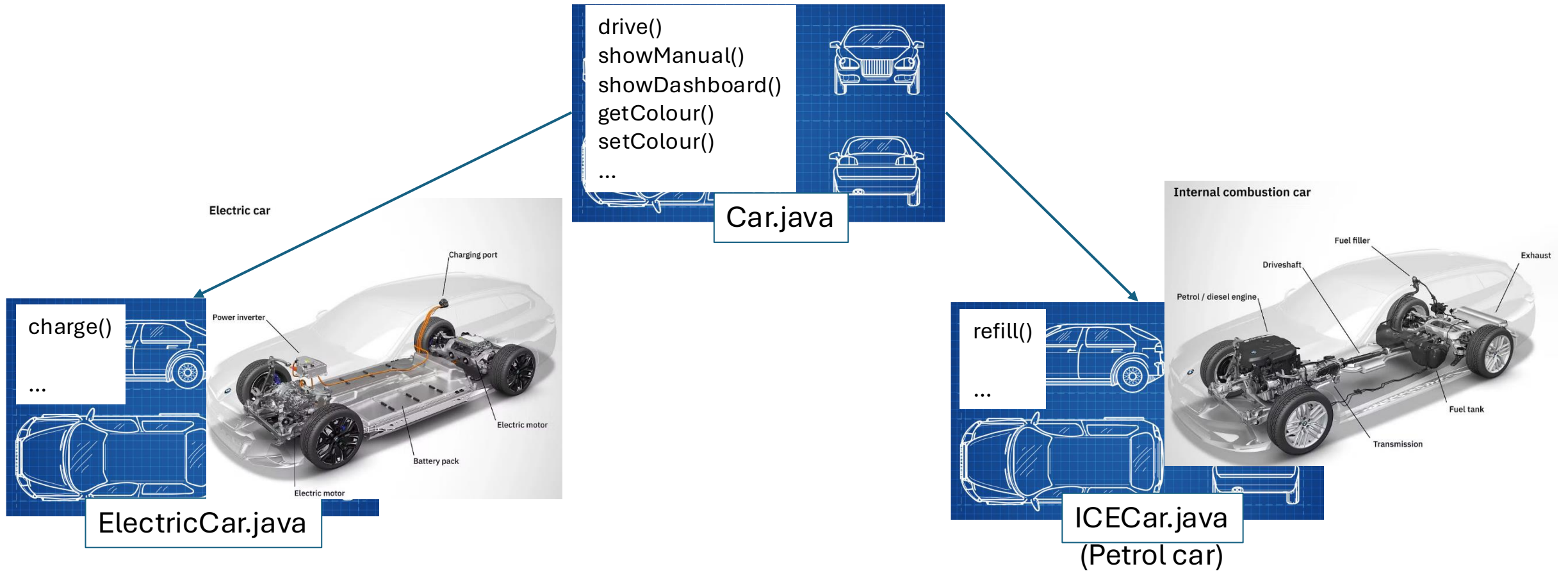
Week 10

Miya

# Your U:PASS Leader - Miya

- 3<sup>rd</sup> years of Bachelor of Information Technology.
- Majoring in Cybersecurity, Networking and Enterprise Software Development.

# What is Inheritance?



- One of Object-Oriented Programming core concept.
- Subclass acquires the properties and methods of a superclass, allowing for **code reuse** and the creation of a **hierarchical relationship** between classes.

# Structure of Inheritance

```
public class Car { Parent Class (Car.java)
    public String manufacturer = "Hyundai Mo

    final public String model;
    public String colour;
    public double energyLevel; // L
    public int odometer; // km
    public int speed;

    public Car(String model, String colour)
        this.model = model;
        this.colour = colour;
        this.energyLevel = 1.0;
        this.speed = 0;
        this.odometer = 0;
    }

    public void drive() {
        System.out.println(model + " starts
```

Inheritance

```
public class ICECar extends Car { Child Class (ICECar.java)
    public String manufacturer = "Hyundai ICE Factory ";
    public ICECar(String model, String colour) { // Interitance
        // Call parent class Car.java's constructor.
        super(model, colour);
    }

    @Override // annotation
    public void drive() {
        // 1. Parent class Car.java's drive()
        super.drive();

        // Overriding
        // 2. Special action of ICECar.java
        if (energyLevel > 0) {
            this.speed = 60;
            this.energyLevel -= 0.1;
            this.odometer += 1;
            System.out.println(x: "Running Internal Combustion Engine! Vroom Vroom!!!");
        } else {
            System.out.println(x: "Not enough fuel. Need to refill.");
        }
    }

    // Unique method for ICECar.java
    public void refill(double liters) {
        this.energyLevel += liters;
        System.out.println(liters + "L refilled.");
        System.out.println("Current Fuel Level: " + this.energyLevel + " L");
    }
}
```

Constructor (Child)

Overridden Method  
(from parent)

Method (Child)

# Back to main

```
import java.util.ArrayList;

public class Main {
    Run | Debug
    public static void main(String[] args){
        ArrayList<Car> garage = new ArrayList<Car>();

        ICECar myCar = new ICECar(model: "myCar ICE ", colour: "Red");

        garage.add(myCar);

        for (int i = 0; i < garage.size(); i++) {
            System.out.println(i + " num : " + garage.get(i).toString());
        }
    }
}
```

# super vs this

- `super.method()` <- call overridden method in the parent class
- `this.method()` <- call method in the current class
- `super.property` <- call property of parent class
- `this.property` <- call property of current class
- `super()` <- call constructor of the parent class
- `this()` <- call another constructor in the same class

# Activity!

- Let's make 'unlimited garage'!
- 1. make garage.
- 2-1. make myCar (ICE) and park at the garage.
- 2-2. make sisCar (Electric) and park at the garage.
- 3. drive myCar. And show dashboard.
- 4. Stop myCar. And show dashboard.
- 5. Drive sisCar. (Check difference!)

# Reminder

- Please keep your paper today!
- Next session will cover week 11.
- Useful website **<https://www.w3schools.com/java/>**

# Any Questions?

- Ask anything!