Programming Fundamentals

Week7

Miya

Your U:PASS Leader - Miya

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

Week 7 Classes – Vocabulary (5 mins)

- **Class** :
- **Object** :
- **Property** :

Week 7 Classes – Vocabulary (5 mins)

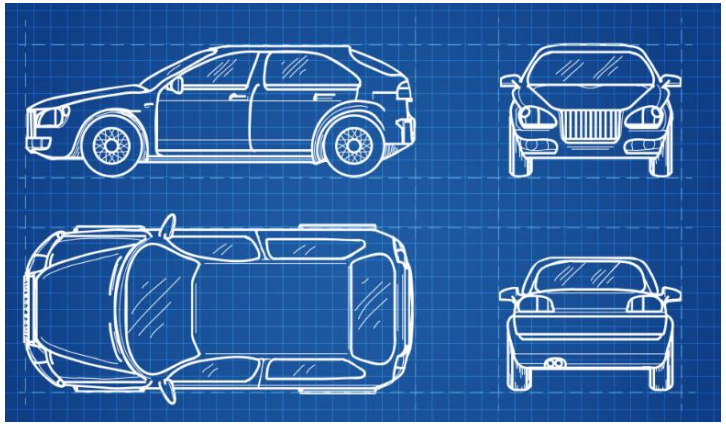
- **Class** : A blueprint or template that defines the structure (properties) and behaviours (methods) of an object.

- **Object** : A specific entity created from a class blueprint that contains i ts own data.

- **Property** : A variable defined by a class that stores the state or data of an object.

What is “Class”?? (5 mins) ???

Manufacturer : Hyundai



Blueprint

Class

Model Name : i30N
Fuel Level : 100

Car

Object/Instance

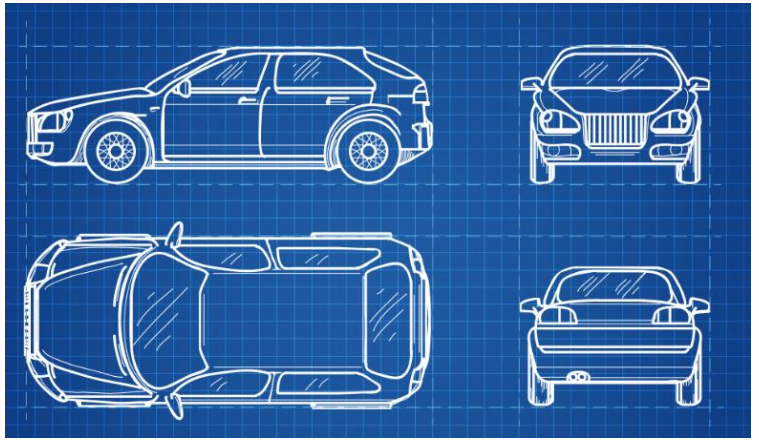
Colour : red, green, purple

Colour

Property

Why Using Class? (5mins)

- “Class” is the core concept of “Object Oriented Programming” (OOP).
- Think why people use Blueprint in the real life.
- **Modularity** -> Easy to maintain
- **Reusability** -> Easy to mass-produce



- **Information Hiding** -> more information at Encapsulation (week 8)

Structure of Class

Class Declaration

File name : Car.java

```java
public class Car {
    static String manufacturer = "Hyundai";

    String model;
    String colour;
    int fuelLevel;

    public Car(String thisModel, String thisColor){
        model = thisModel;
        colour = thisColor;
        fuelLevel = 30;
    }

    public void drive(){
        if (fuelLevel > 0){
            System.out.println(x: "Drive!");
            fuelLevel -= 10;
        } else {
            System.out.println(x: "Out of fuel!");
        }
    }

    public void refillFuel(int n){
        fuelLevel += n;
    }
}
```

Global Property

Local properties

Constructor

- Default factory setup
- Same name with class

Methods

Behaviours / Actions

Class

Blueprint

Back to the “Main”

File name : Main.java

“main” Method (not Constructor)

Making Instance from Class (Object Instantiation)

Making Car from Blueprint

Activity : Let's play around with the Car!

Folder name : Car <- Car.java, Main.java (Should contain in the same folder)

```java
public class Car {
    static String manufacturer = "Hyundai";

    String model;
    String colour;
    int fuelLevel;

    public Car(String thisModel, String thisColor){
        model = thisModel;
        colour = thisColor;
        fuelLevel = 30;
    }

    public void drive(){
        if (fuelLevel > 0){
            System.out.println(x: "Drive!");
            fuelLevel -= 10;
        } else {
            System.out.println(x: "Out of fuel!");
        }
    }

    public void refillFuel(int n){
        fuelLevel += n;
    }
}
```

```java
public class Main {
    Run | Debug
    public static void main(String[] args){
        Car myCar = new Car(thisModel: "i30N", thisColor: "Blue");
    }
}
```

1 : Drive until fuel run out! And refill the fuel.
2 : Add 1 more method (behaviour/action)
3 : Add 1 more local property
4 : Add 1 more global property
5 : Try to change colour. I don't like this!
6 : Try to change manufacturer…?
7 : My sister wants new car. Can you make new one for her?

Reminder

- Please keep your paper today!
- Next week, StuVac! So, no session.
- Coming session, will cover week 8 – Encapsulation!
- Useful website **https://www.w3schools.com/java/**

Any Questions?

- Ask anything!