Introduction to Java week#5

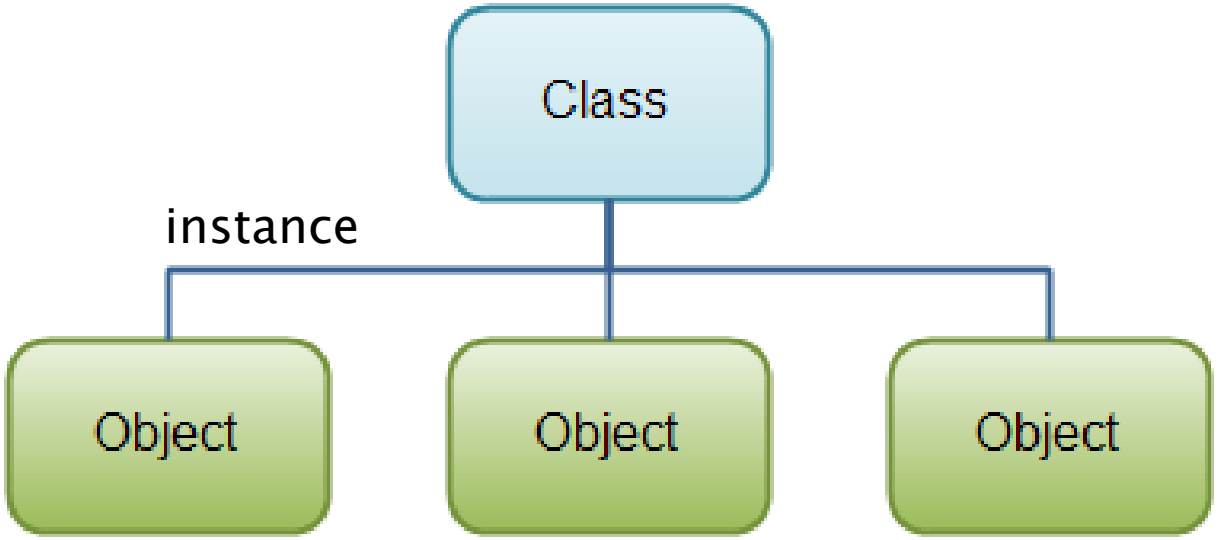
09/06/2023

week	Topic	Calendar
1	JAVA IDE (NetBean) Installation ,Configuration and Compile	3 - 7 Apil 2023
2	Basic structure of Java ,Data & Variable type, operator & basic logic	17 - 21 Apil 2023
3	Function(Method) create & calling, Input & output	20 - 24 Apil 2023
4	Loop statement ,Array variable	27 - 31 Apil 2023
5	Object-oriented programming (OOP),Class & Object, Encapsulation	1 - 5 May 2023
6	Inheritance, Polymorphism, Interfaces	8 - 12 May 2023
7	Packages, Access Modifiers(Public ,Protected ,Private class)	15 - 19 May 2023
8	Collections (Array list, HashMap, Stack)	22 - 26 May 2023
9	Exception	29 May - 2 June 2023
10	Woking with files(Read, Write)	5 - 9 June 2023
11	Thread Programing	12 - 16 June 2023

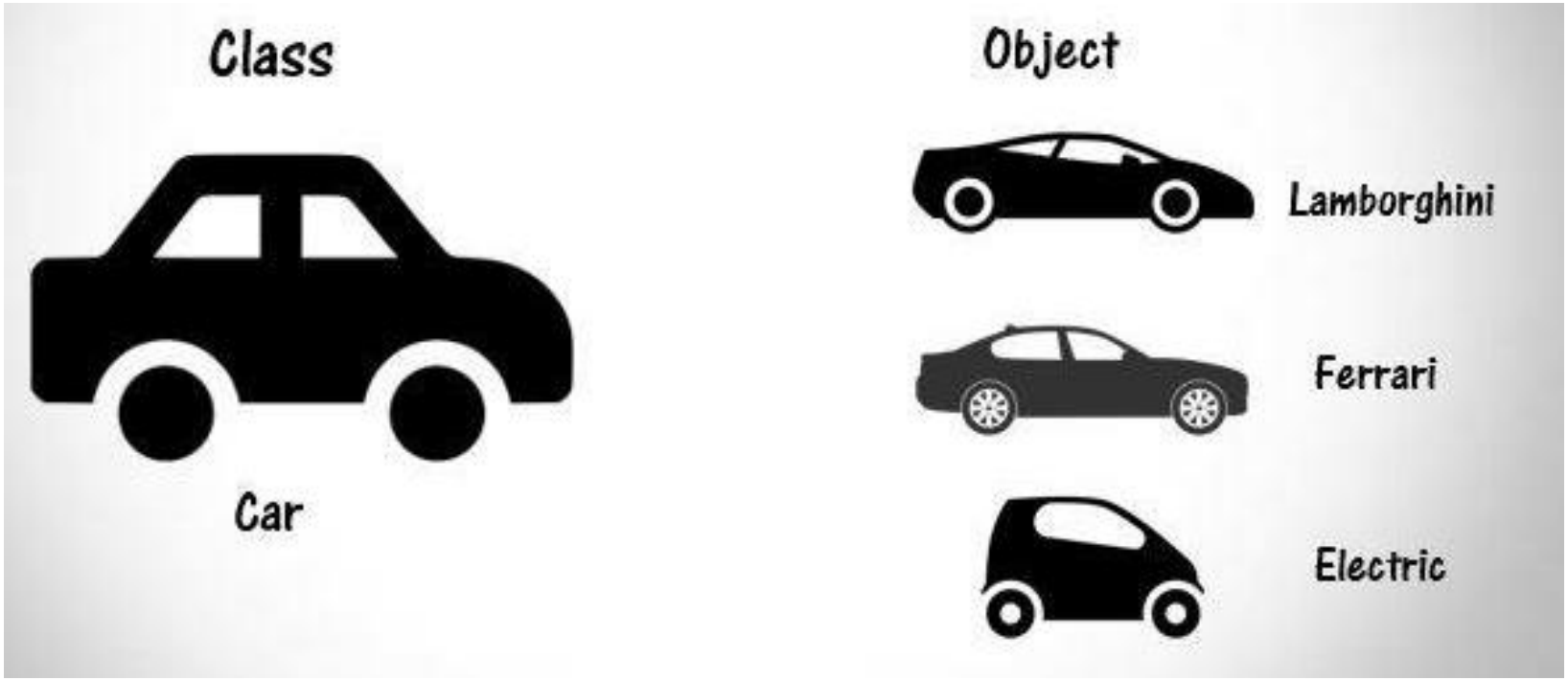
Object-Oriented Programing(OOP)

What is OOP?

is a programming paradigm based on the concept of "objects", which can contain data and code. The data is in the form of fields (often known as attributes or properties), and the code is in the form of procedures (often known as methods).



Object-Oriented Programing(OOP)



Object-Oriented Programing(OOP)

Four Principles of OOP

The four pillars of object-oriented programming are

- **Encapsulation**:

  containing information in an object, exposing only selected information

- **Inheritance**:

  child classes inherit data and behaviors from the parent class

- **Abstraction**:

  only exposing high-level public methods for accessing an object

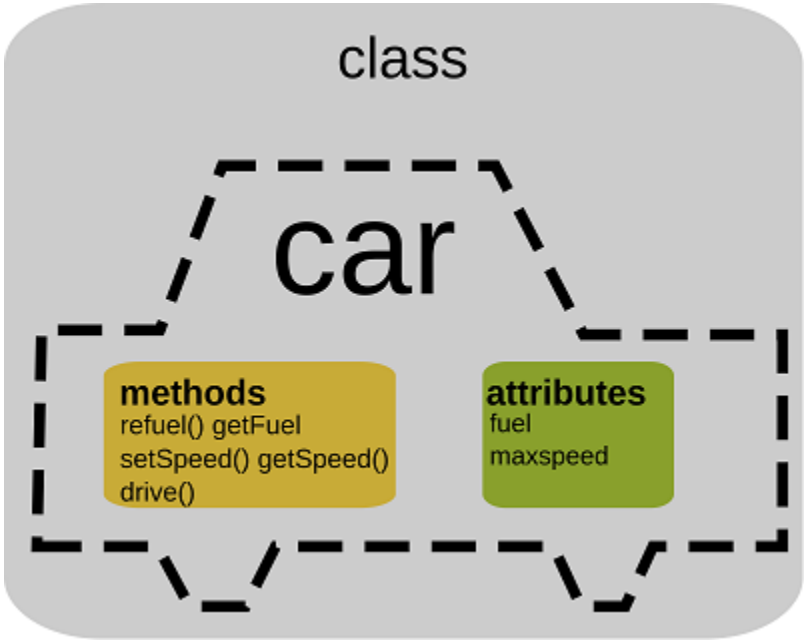
- **Polymorphism**:

  many methods can do the same task

Class and Object

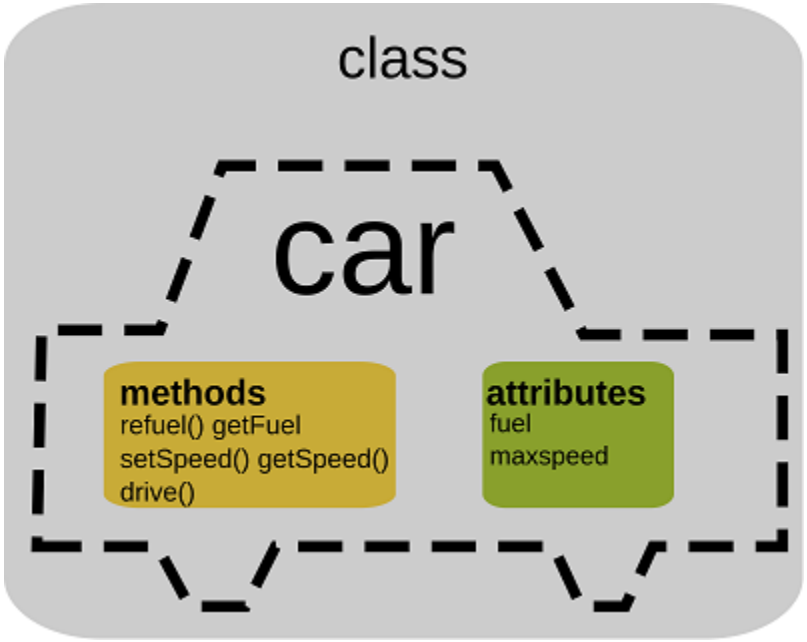
Class and Object

Class design



Class and Object

Class design




```
public class Car {
    public float fuel;
    public int maxSpeed;
    public int currentSpeed = 0;
    public String color;

    public void reFuel() {
        fuel = 100;
    }
    public float getFuel() {
        return fuel;
    }
    private void setSpeed(Int speed) {
        currentSpeed = speed;
    }
    public void drive() {
        setSpeed(currentSpeed + 1);
        if(currentSpeed >= maxSpeed) {
                currentSpeed = maxSpeed;
        }
    }
}
```

Class and Object

Class creation

```
class ClassName {
    // member variables
    // member methods
}
```

Class Example

```
class Car{
    public String name;
    public float fuel;
    public float speed = 4.0f;

    void setSpeed (float newSpeed) {
        speed = newSpeed;
    }

    float getSpeed () {
        return speed;
    }

    void drive () {
        System.out.println(name + " is running at speed " + speed + "
    km/s.");
    }
}
```

Class Example

```
public class ObjectExample {
    public static void main(String[] args) {
        Car c1 = new Car();
        Car c2 = new Car();
        c1.name = “Ferrari";
        c1.fuel = 20;
        c1.speed = 3.5f;
        c2.name = “Lamboghini";
        c2. fuel = 25;
        System.out.println(“c1 is " + c1.name);
        System.out.println(“It fuel is " + c1.fuel + “ lites.");
        System.out.println(c1.name + " has speed " + c1.speed + " km/s.");
        c1.drive();
        System.out.println("\nc2 is " + c2.name);
        System.out.println(“It fuel is " + c2.fuel + " lites.");
        System.out.println(c2.name + " has speed " + c2.getSpeed() + " km/s.");
        c2.drive();
        c2.setSpeed(5.0f);
        c2.drive();
    }
}
```

Class Constructor

- a fundamental part of object-oriented programming. They allow you to create and properly initialize objects of a given class, making those objects ready to use.

```
class Car{
    public String name;
    public float fuel;
    public float speed = 4.0f;

    public  Car (String name, float fuel)
    {
        this.name = name;
        this.fuel = fuel;
    }
```

```
Car c1 = new Car("Ferrari", 20);
Car c2 = new Car("Lamboghini", 25);
```

Overloading Constructor

```
class Car{
    public String name;
    public float fuel;
    float speed = 4.0f;

    public  Car (String name, float fuel)
    {
        this.name = name;
        this.fuel = fuel;
    }
    public  Car (String name, float fuel, speed speed)
    {
        this.name = name;
        this.fuel = fuel;
        this.speed = speed;
    }
```

```
Car c1 = new Car("Ferrari", 20);
Car c2 = new Car("Lamboghini", 25, 100);
```

Static Member

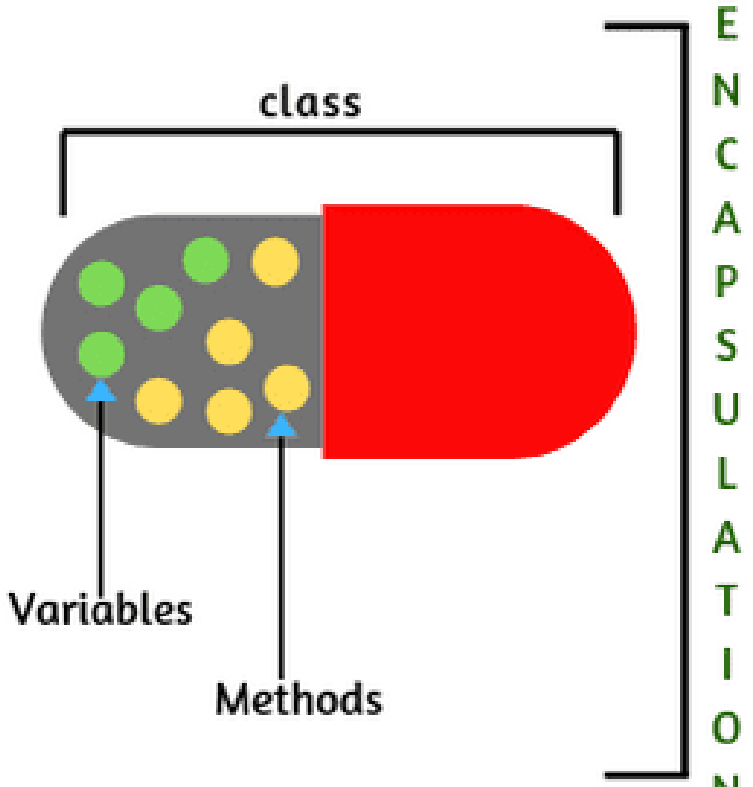
```
class StaticMember {

   public static String NAME_PREFIX = "Mr.";

   public static void displayName (String name) {

   System.out.println(NAME_PREFIX + " " + name);

   }
}
```

```
System.out.println(StaticMember.NAME_PREFIX + " " + c1.name);

System.out.println("Age: " + c2.fuel); StaticMember.displayName(c2.name);
```

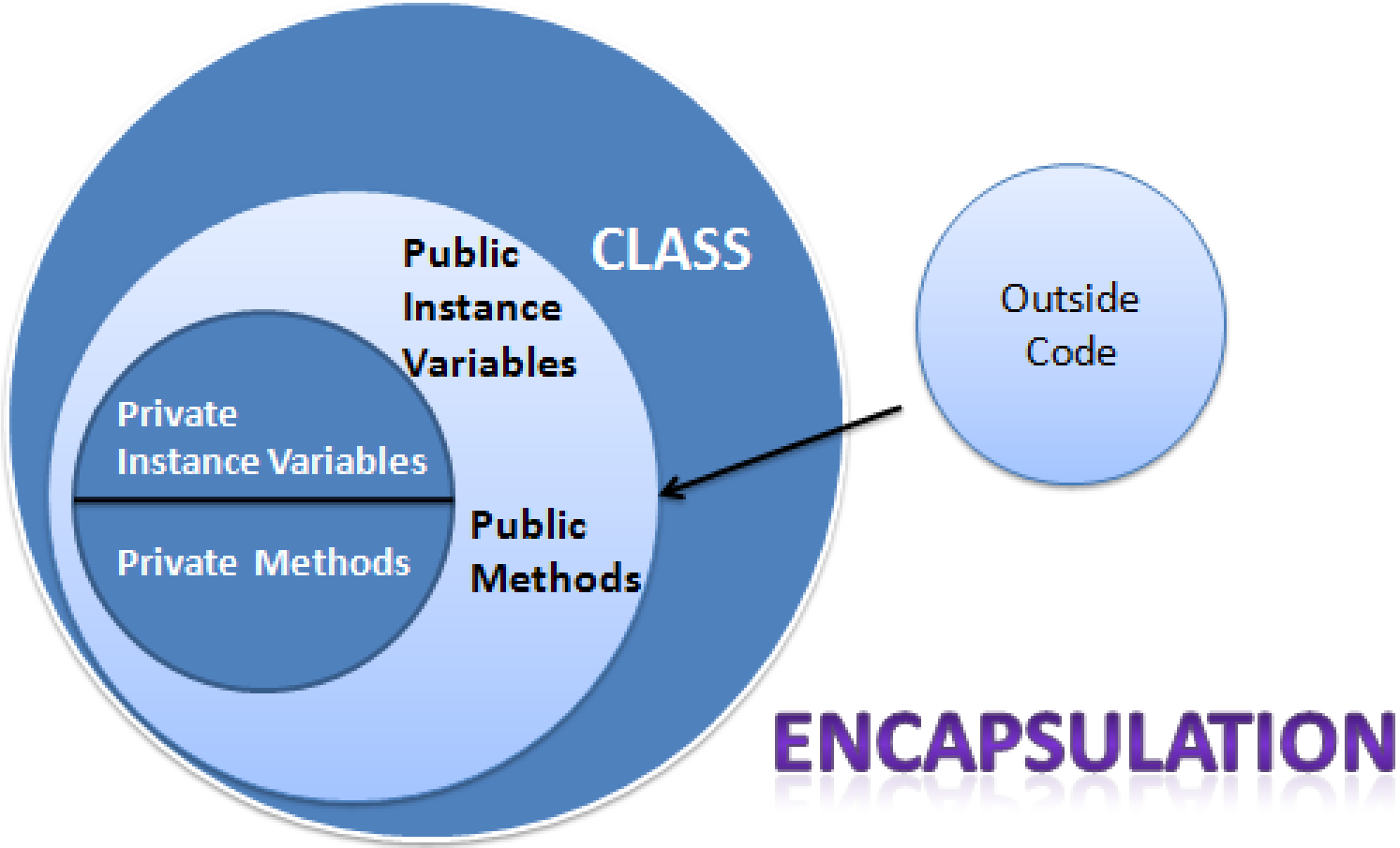
Encapsulation

```
class
{

   data members
         +
   methods (behavior)

}
```

ENCAPSULATION



Encapsulation

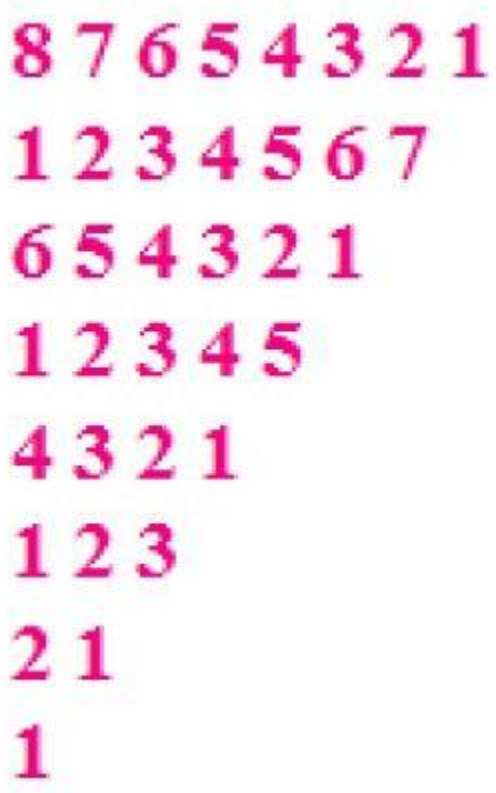


Encapsulation

```
class Car{
    public String name;
    public float fuel;
    private float speed = 4.0f;

    void setSpeed (float newSpeed) {
        speed = newSpeed;
    }

    float getSpeed () {
        return speed;
    }

    void drive () {
        System.out.println(name + " is running at speed " + speed + "
    km/s.");
    }
}
```

Assignments

สร้าง loop program ที่แสดงผลดังรูป



Assignments

ออกแบบ และ สร้างคลาส สำหรับ “คน” ให้มีคุณสมบัติ ชื่อ, อายุ , เพศ, อาชีพ และ มีเมธอด แนะนำตัว, กิน, นอน, เดิน และทำการ สร้างออปเจ็คจากคลาสที่สร้างเป็น คน 3 คนที่แตกต่างกัน

Thank you