

Camel case → Class name → Starts with Upper Case, + new word starts with Upper Case  
 Variable, method → Starts with lower case

Car Name

Upper case  
variable, method → starts with lower  
case +  
new word starts with  
uppercase

```

1 public class Car {
2     private String carName;
3     private String fuelType;
4     private double engineCapacity;
5     private double mileage;
6     private double power;
7
8     public Car() {
9     }
10
11     public Car(String carName, String fuelType, double engineCapacity, double mileage, double power) {
12         this.carName = carName;
13         this.fuelType = fuelType;
14         this.engineCapacity = engineCapacity;
15         this.mileage = mileage;
16         this.power = power;
17     }
18
19     public void accelerate() {
20         System.out.println(carName + " Car is accelerating...");
21     }
22
23     public void applyBrakes() {
24         System.out.println(carName + " Car is slowing/stopping...");
25     }
26
27     public void changeGears() {
28         System.out.println("Shifting the Gears for Car " + carName);
29     }
30 }
```

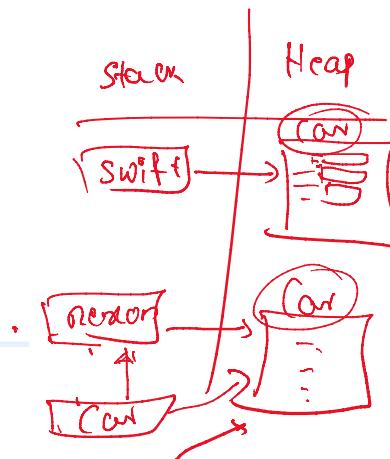
private ✓ protected ✓  
default ✓ public ✓

format → same for all instance of class

```

1 public class CarDemo {
2
3     public static void main(String[] args) {
4
5         Car swift = new Car("Suzuki Swift", "Petrol", 1198.6, 23.2, 96.3);
6         swift.accelerate();
7         swift.changeGears();
8         swift.steer();
9         swift.applyBrakes();
10        System.out.println();
11
12        Car nexon = new Car("Tata Nexon", "Diesel", 1496.8, 17.6, 125.6);
13        nexon.accelerate();
14        nexon.changeGears();
15        nexon.steer();
16        nexon.applyBrakes();
17    }
18 }
```

Car car = nexon;



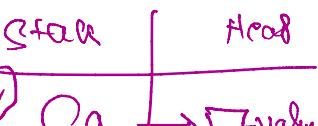
hashCode(): → hash value  
unique value → object

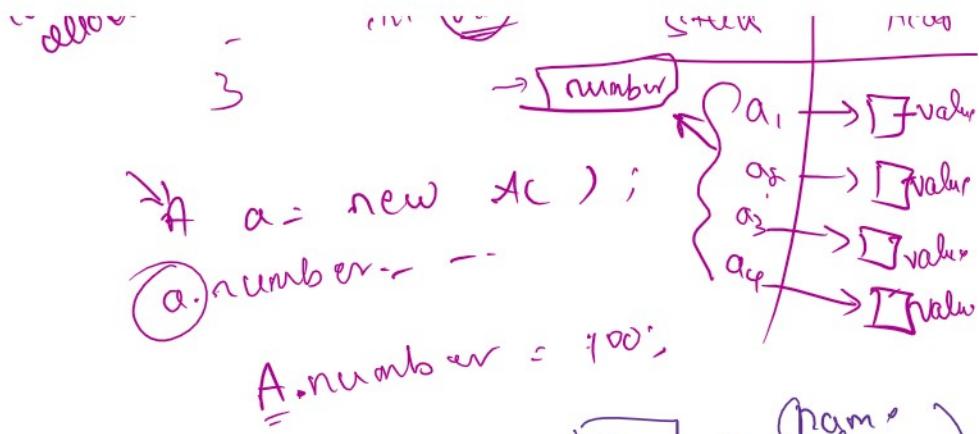
static ← variable  
method  
↳ class level variable / method.

Class A {  
common and  
dilogoic static int number;  
int value;  
}

→ number on

A as=new().  
a1  
a2  
a3  
a4  
i.





class SUV extends Car {  
 driveMode  
 sunroof

}

- \* Create SUV class extending Car class.  
 → add 2 attributes
  - driveMode (String)
    - Eco
    - Sports
    - City
  - sunroof (boolean)
    - on
    - false
- \* Create SUV car object & init  
 all attributes.

- \* Create SUV car object & print
- \* display all features.

- \* add a method SUV, to change
  - drivemode (center)
- power change
  - if ECO → power 73.2 bhp
  - city → 92.8 bhp
  - sports → 110.3 bhp

