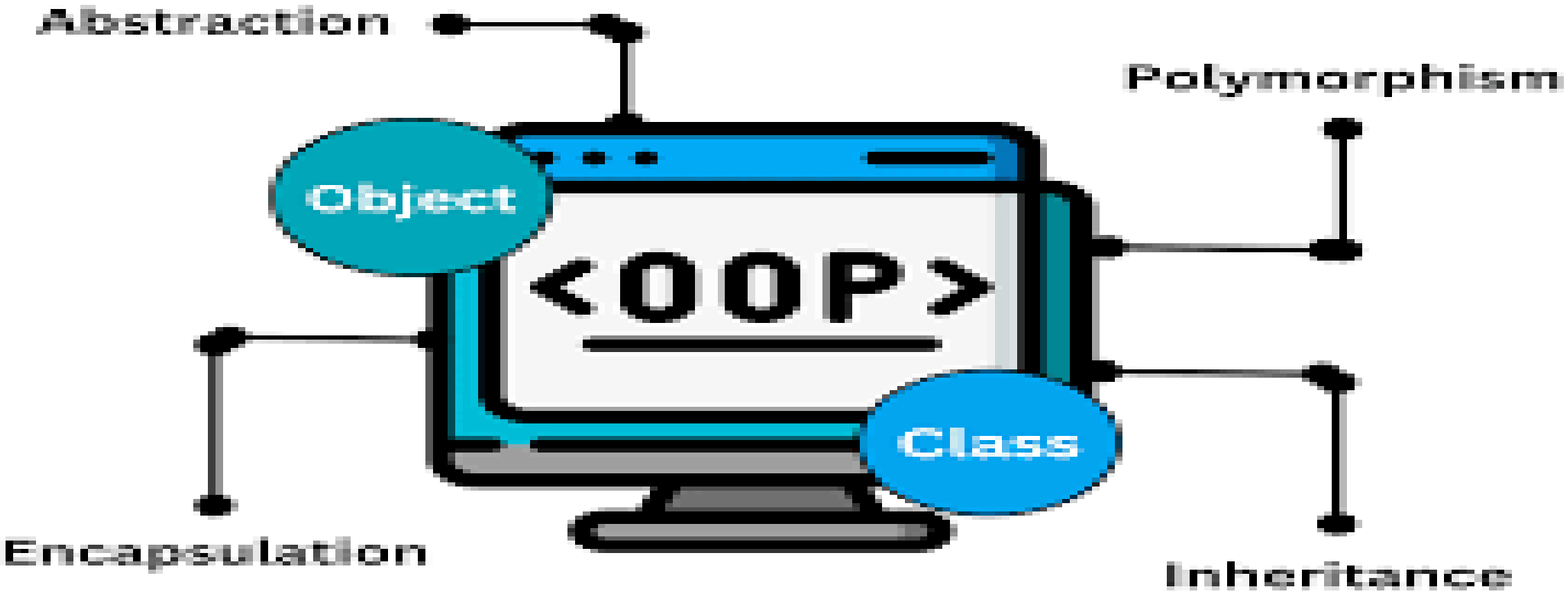
Object Oriented Programming with Java (OOPJ)

Session 3: Operators & Basics

Kiran Waghmare


Abstraction
Polymorphism
Object
<OOP>
Class
Encapsulation
Inheritance

class

car

methods

refuel() getFuel

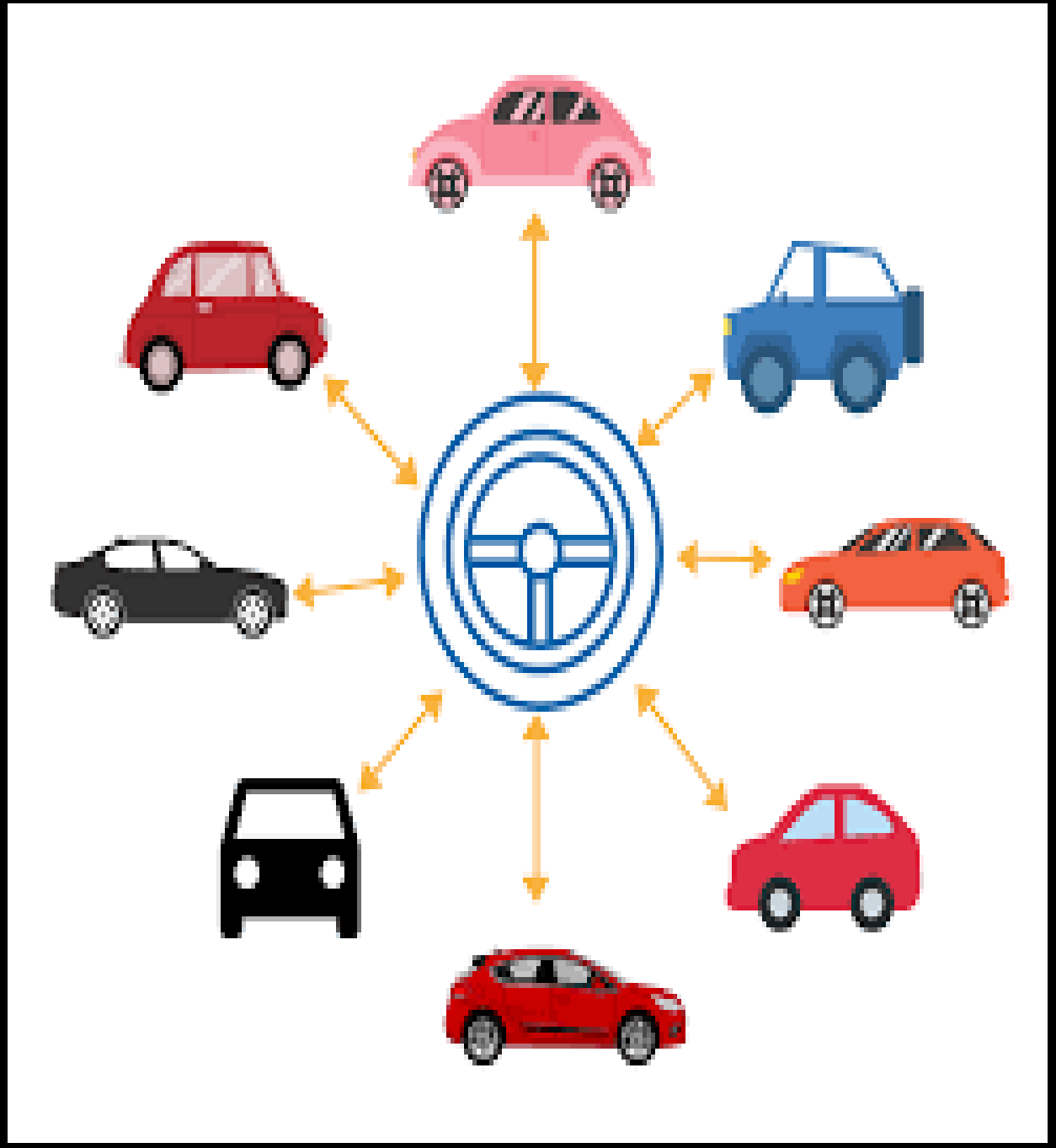
setSpeed() getSpeed()

drive()

attributes

fuel

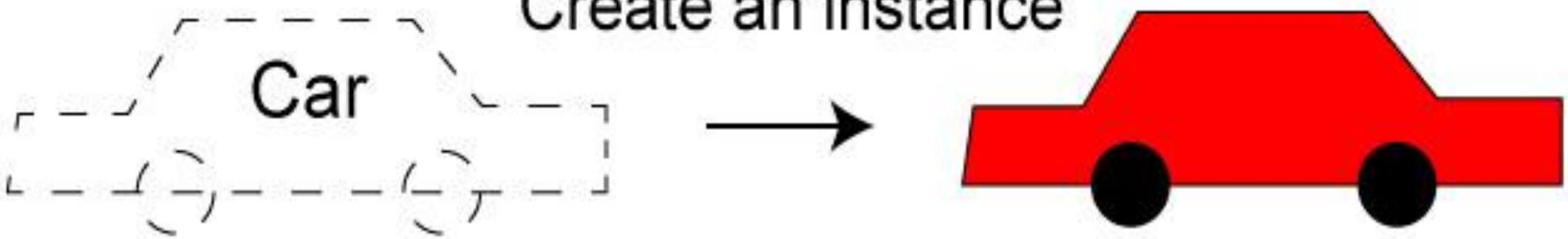
maxspeed



Class

Object

Create an instance



Properties	**Methods** - behaviors
color	start()
price	backward()
km	forward()
model	stop()

Property values	**Methods**
color: red	start()
price: 23,000	backward()
km: 1,200	forward()
model: Audi	stop()

Examples of Objects

LightBulb

- **state/attributes**
  - on (true or false)
- **behavior**
  - switch on
  - switch off
  - check if on

Car

- **state/attributes**
  - # of liters of gas in tank
  - total # of km run so far
  - efficiency (km/liter)
- **behavior**
  - drive
  - load gas
  - change efficiency
  - check gas
  - check odometer reading

BankAccount

- **state/attributes**
  - balance
- **behavior**
  - deposit
  - withdraw
  - check balance

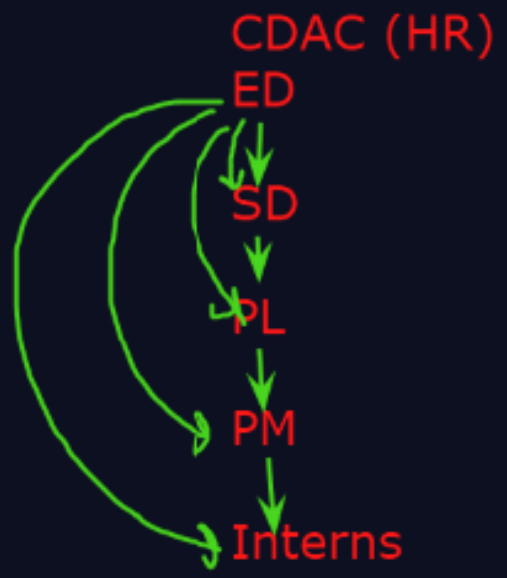
Note

- **each object is an "instance" of that "class" of object**
- **each instance has its own values for its attributes**
  - e.g., different accounts can have different balances

OOPS : Objected Oriented Programming:

-1. Modularity : Divides the program into objects.
-2. Resusability : Inherit existing functionality.
-3. Scalability : Easier to manage large appications.
-4. Secure : Abstraction and Encapsulation we restrict direct access to data

Data hiding : Access Modifiers (public,private, protected, defa



Real world Example:

-1. Class
-2. Object
-3. Methods

Key Features:

-1. Class
-2. Object
-3. Abstraction
-4. Encapsulation
-5. Inheritance
-6. Polymorphism

4 pillars → Major Pillars:Abstraction, Encapsulation, Modularity, Hiera

4 pillars → Minor Pillars:Typing, Concurrency, Persistence

Real world Example:

-1. Class
-2. Object
-3. Methods

Key Features:

-1. Class
-2. Object
    -Objects are real worlds entity, and classes are their blueprints.

```
class Student{
    int i=10;
    void display(){
    }
    p.s.v.main(){

    }
}
```

```
        show();//Function call
        SOP (show());//Direct printing od return data
        //or
        String x = show();
        SOP(x);
    }
}
```

095_Rohan Sharma_KH raised hand View ×

Reference:
-A reference variable stores the memory address of an object.

-It is used to access the object'fields and methods.

Syntax:

```
    <Classname> <refernce name> = new <class constructor name>;
```

```
Employee e1 = new Employee();
```