Object Oriented Programming

Eli

Why?

An ***algorithm*** is a step-by-step process.

A ***computer program*** is a step-by-step set of instructions for a computer.

Every computer program is an algorithm.

History of Computer Programming

How do we see ourselves in this world ???

Programming Languages

Machine, Assembly & High level

Machine Languages

1 and 0

Machine Languages- Example

```
1110100010101     111010101110
10111010110100    10100011110111
```

Assembly Languages

Bit easier

Assembly Languages-Example

```
ADD 1001010, 1011010
```

High Level languages

```
grossPay = basePay + overTimePay
```

High Level languages - Groups

Procedural languages

Object Oriented languages (OOP)

Procedural languages

Sequential sets of linear commands

C, LISP, Fortran, HTML, Perl

Object Oriented languages

Focus: NOT on structure, on modeling data

C++, Java

Object Oriented languages

Programmers code using “blueprints” of data models called classes.

Object Oriented languages

Programmers code using “blueprints” of data models called classes.

What?

Object Oriented Programming (OOP)

Everything in OOP is grouped as self sustainable "objects"

Object Oriented Programming (OOP)

Define not only the data type of a data structure, but also the types of operations/methods (functions) that can be applied to the data structure.

Object Oriented Programming (OOP)

In this way, the data structure becomes an object that includes both data and functions (methods) in one unit.

Object Oriented Programming (OOP)

In addition, programmers can create relationships between one object and another.

Object Oriented Programming (OOP) - Key idea

The real world can be “accurately” described as a collection of objects that interact.

OOP Terminology

<u>Object</u>

- usually a person, place or thing (a noun)

<u>Method</u>

- an action performed by an object (a verb)

OOP Terminology

Property or attribute

- Characteristics of certain object.

Class

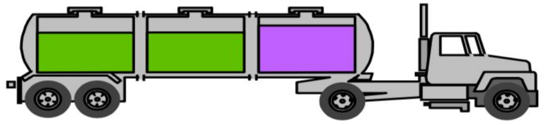
- a category of similar objects (such as automobiles), does not hold any values of the object’s attributes/properties

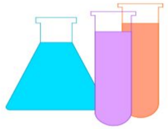
Everything in this world is an object

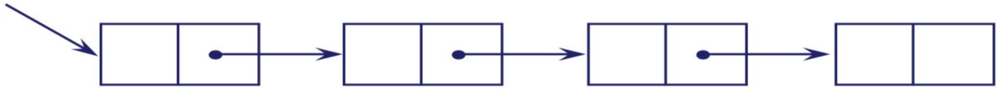
A flower, a tree, an animal
- A student, a professor
- A desk, a chair, a classroom, a building
- A university, a city, a country
- The world, the universe
- A subject such as CS, IS, Math, History, …
- An information system, financial, legal, etc..

Object

- Informally, an object represents an entity, either physical, conceptual, or software.
  - Physical entity
  - Conceptual entity
  - Software entity


Truck


Chemical Process


Linked List

Object - Lets define it formally

An object is a computational entity that:

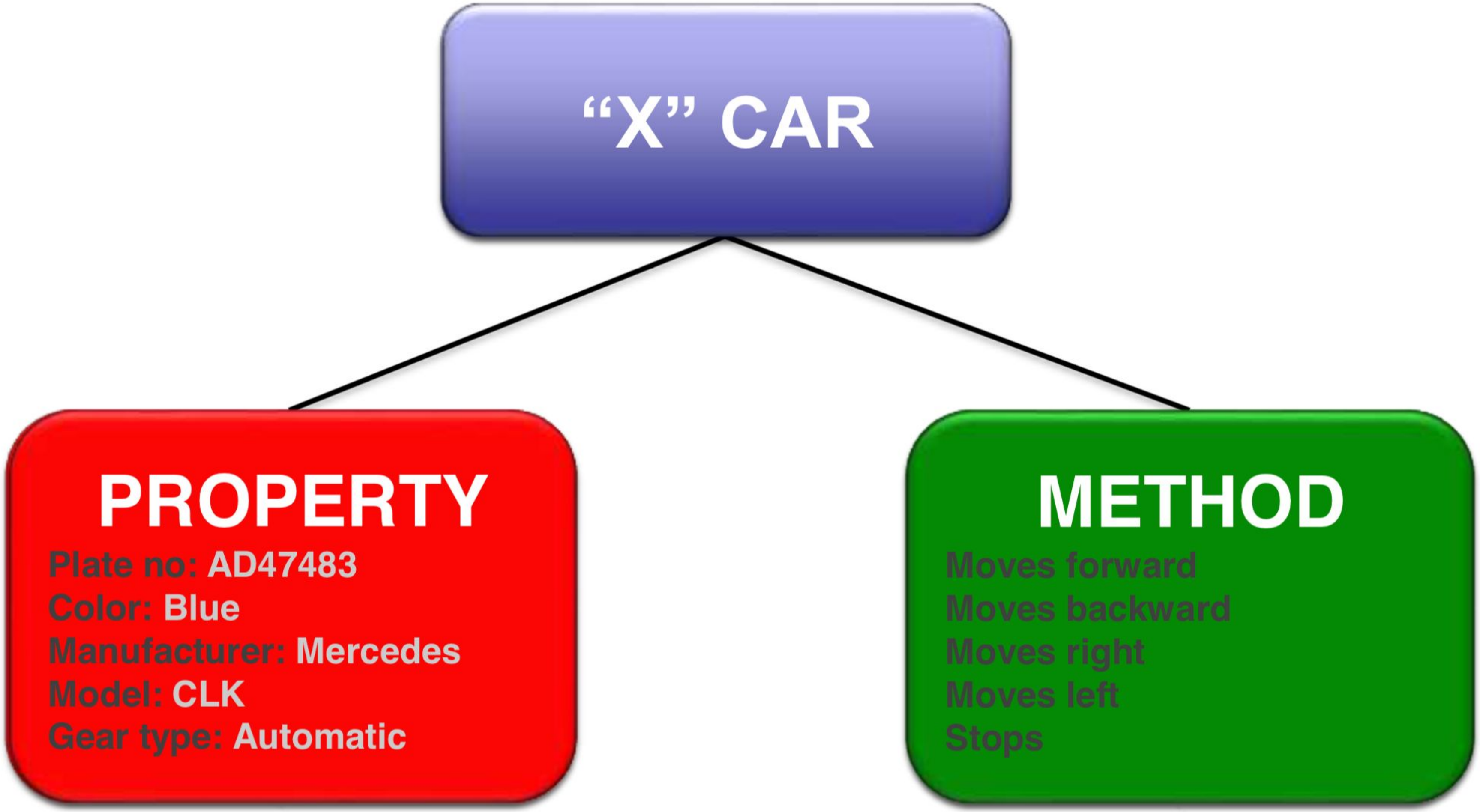
1. Encapsulates some state

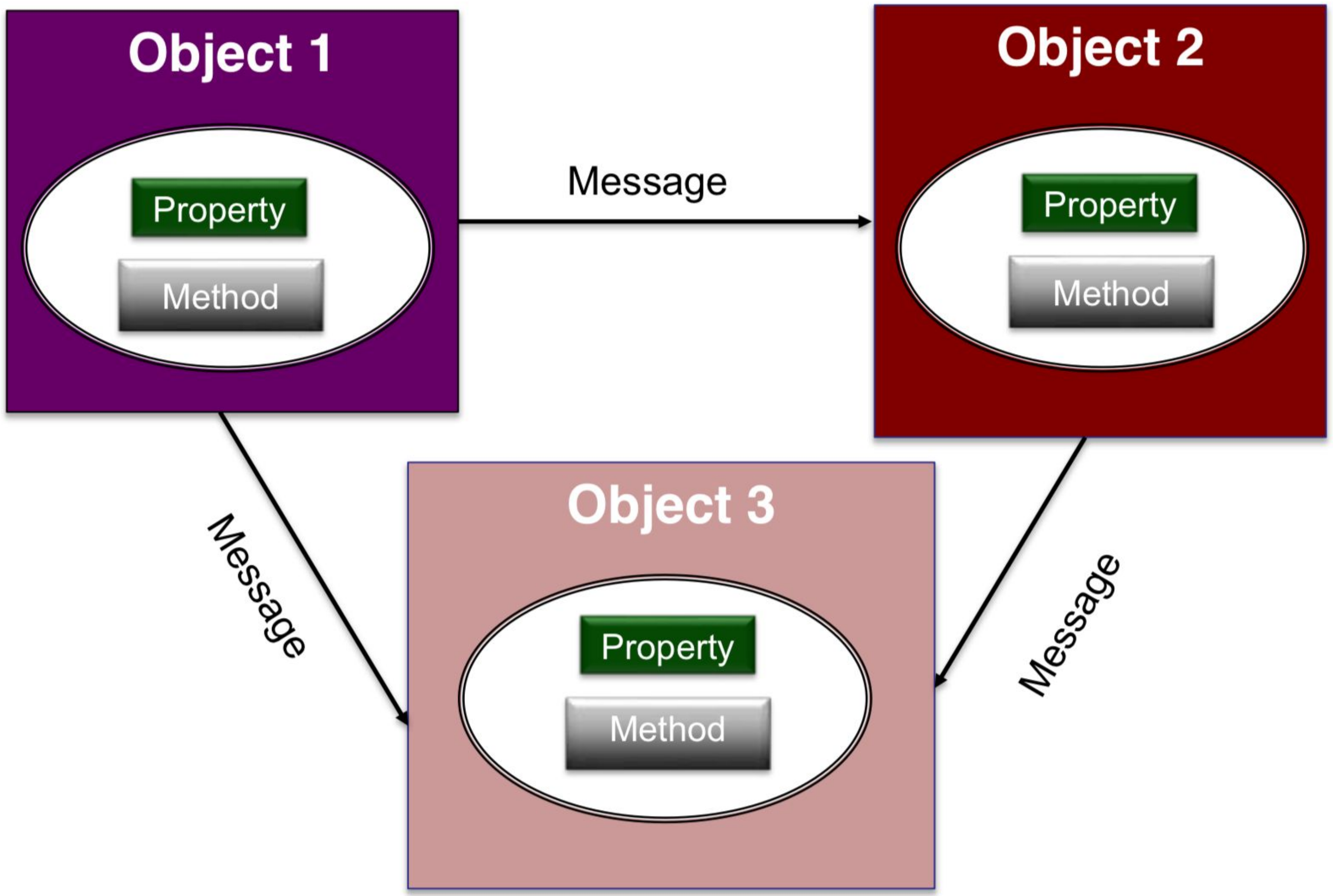
Object - Lets define it formally

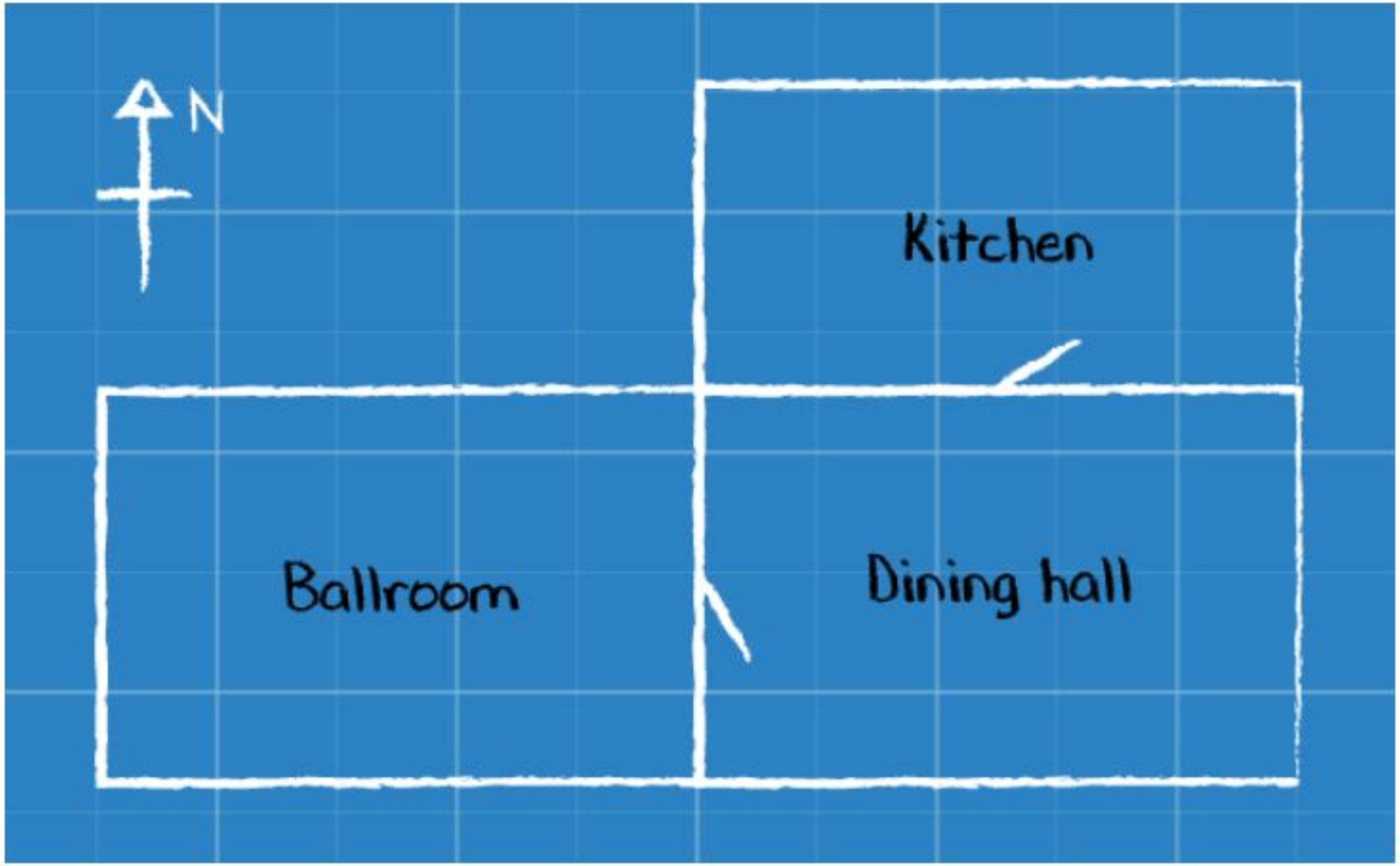
2. Is able to perform actions, or methods, on this state

Object - Lets define it formally

3. Communicates with other objects via message passing


"X" CAR
PROPERTY
Plate no: AD47483
Color: Blue
Manufacturer: Mercedes
Model: CLK
Gear type: Automatic
METHOD
Moves forward
Moves backward
Moves right
Moves left
Stops


Object 1
Property
Method
Message
Object 2
Property
Method
Message
Object 3
Property
Method
Message


N
Kitchen
Ballroom
Dining hall

Object & Class

An Object is a Class when it comes alive!

Object & Class

Homo Sapien is a class, *Jill and Jack* are objects

Object & Class

Animal is a class *“Snowball”* the cat is an object

Object & Class

Vehicle is a class *My neighbor's SAAB i*s an object

CLASS	OBJECT
Class is a data type	Object is an instance of Class.
It generates OBJECTS	It gives life to CLASS
Does not occupy memory location	It occupies memory location.
It cannot be manipulated because it is not available in memory *(except static class)*	It can be manipulated.

Object is a class in *“runtime”*

Objects need to collaborate

Objects are useless unless they can collaborate together to solve a problem

Objects need to collaborate

Each object is responsible for its own behavior and status.

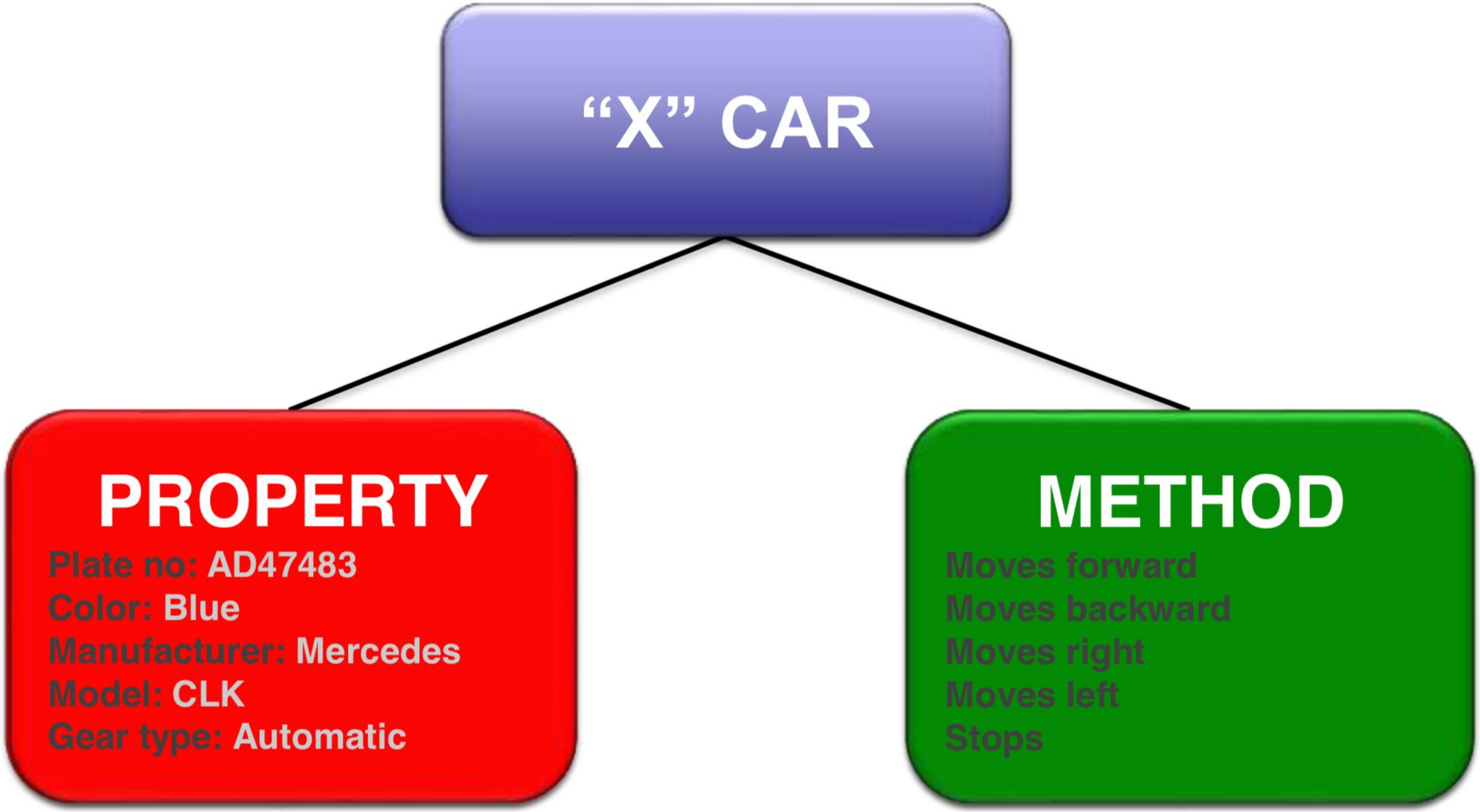
Objects need to collaborate

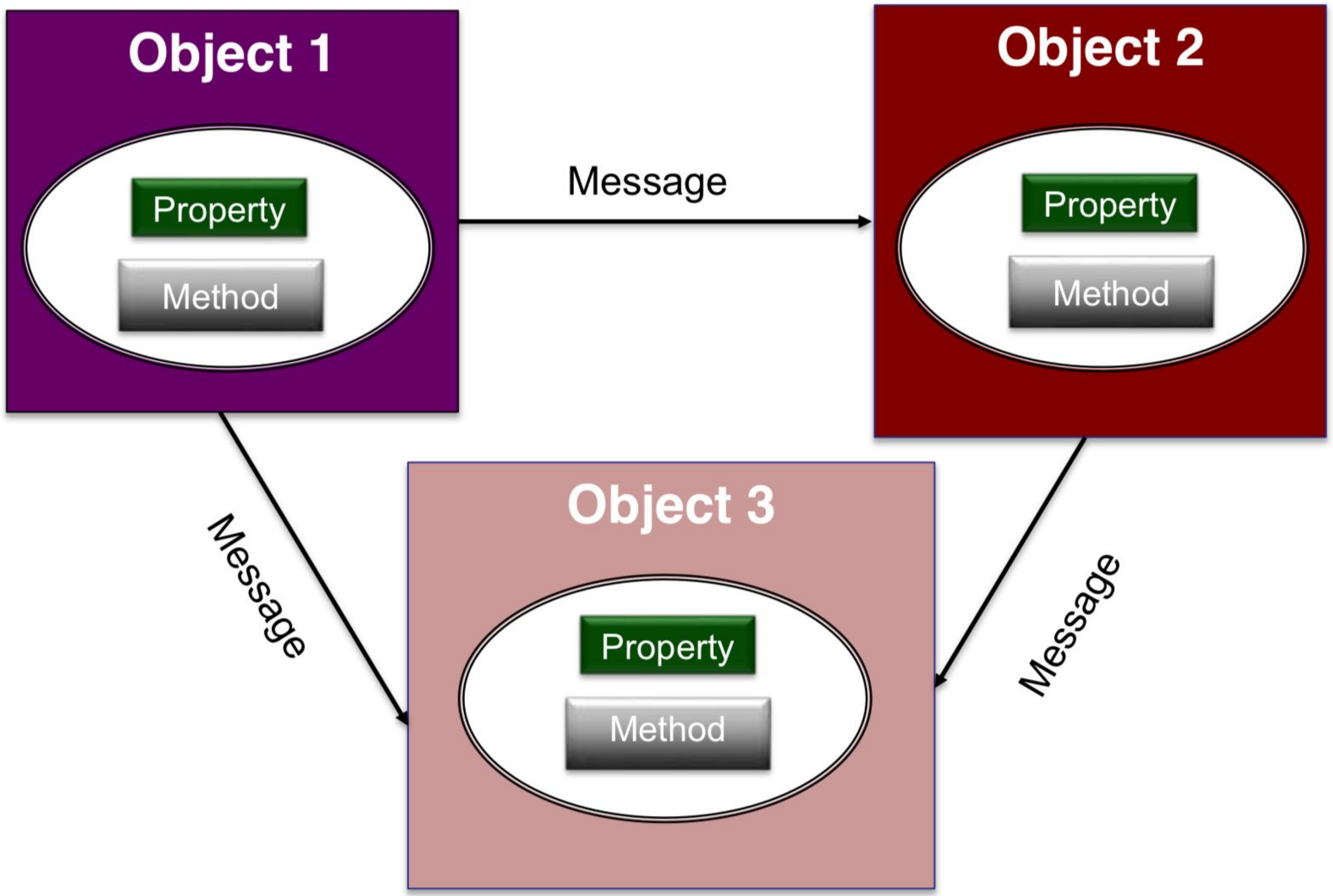
No one object can carry out every responsibility on its own.

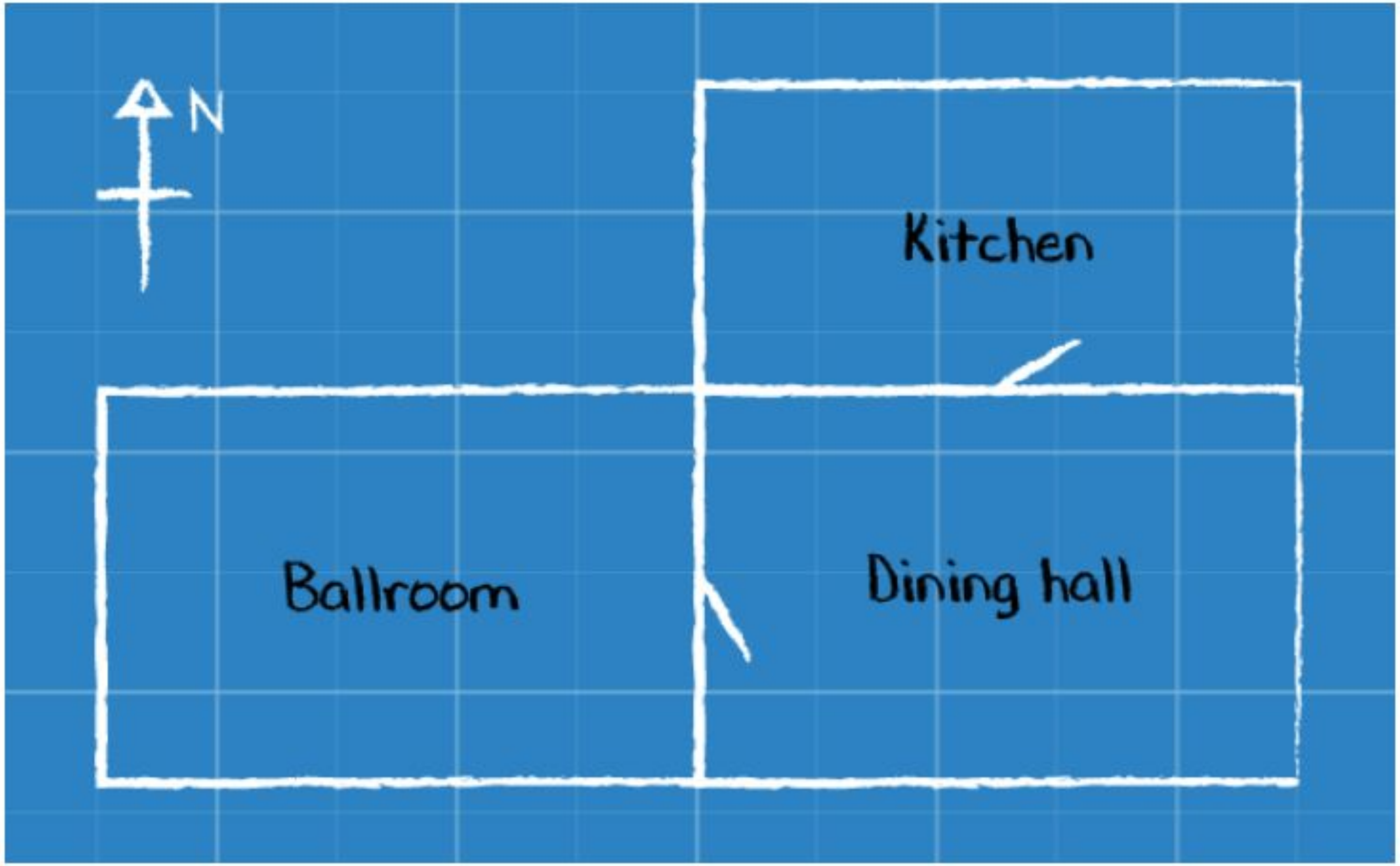
How?

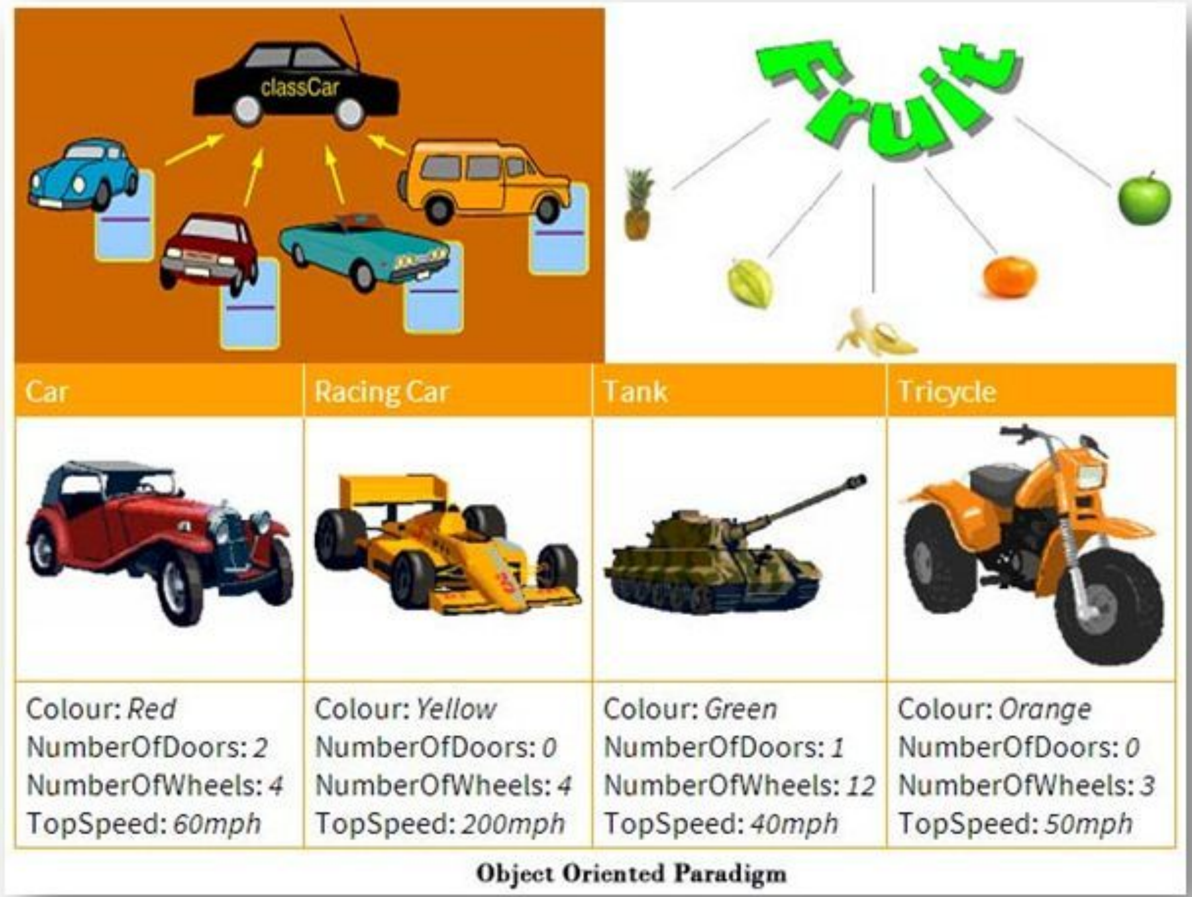
How do objects interact with each other?

They interact through messages


"X" CAR
PROPERTY
Plate no: AD47483
Color: Blue
Manufacturer: Mercedes
Model: CLK
Gear type: Automatic
METHOD
Moves forward
Moves backward
Moves right
Moves left
Stops

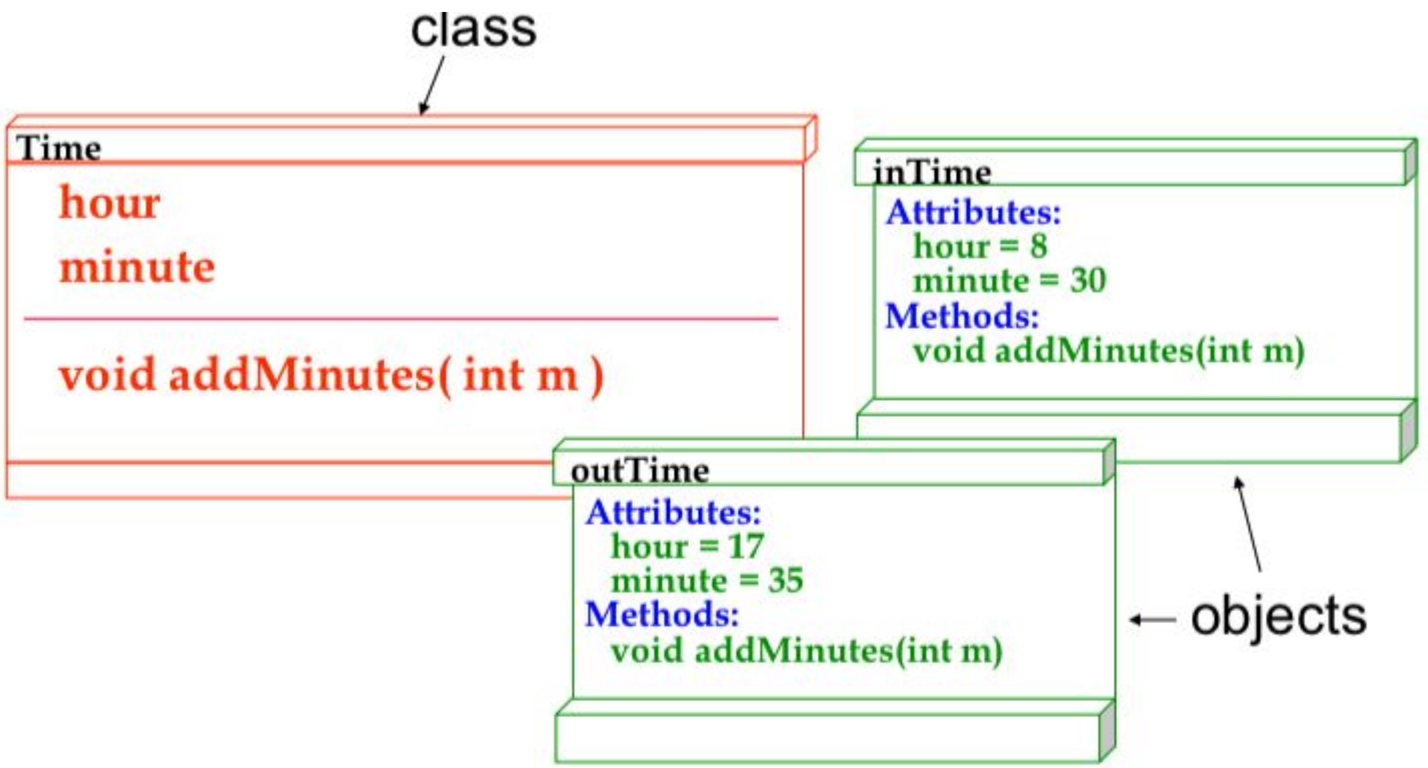

Object 1
Property
Method
Message
Object 2
Property
Method
Message
Object 3
Property
Method
Message


N
Kitchen
Ballroom
Dining hall



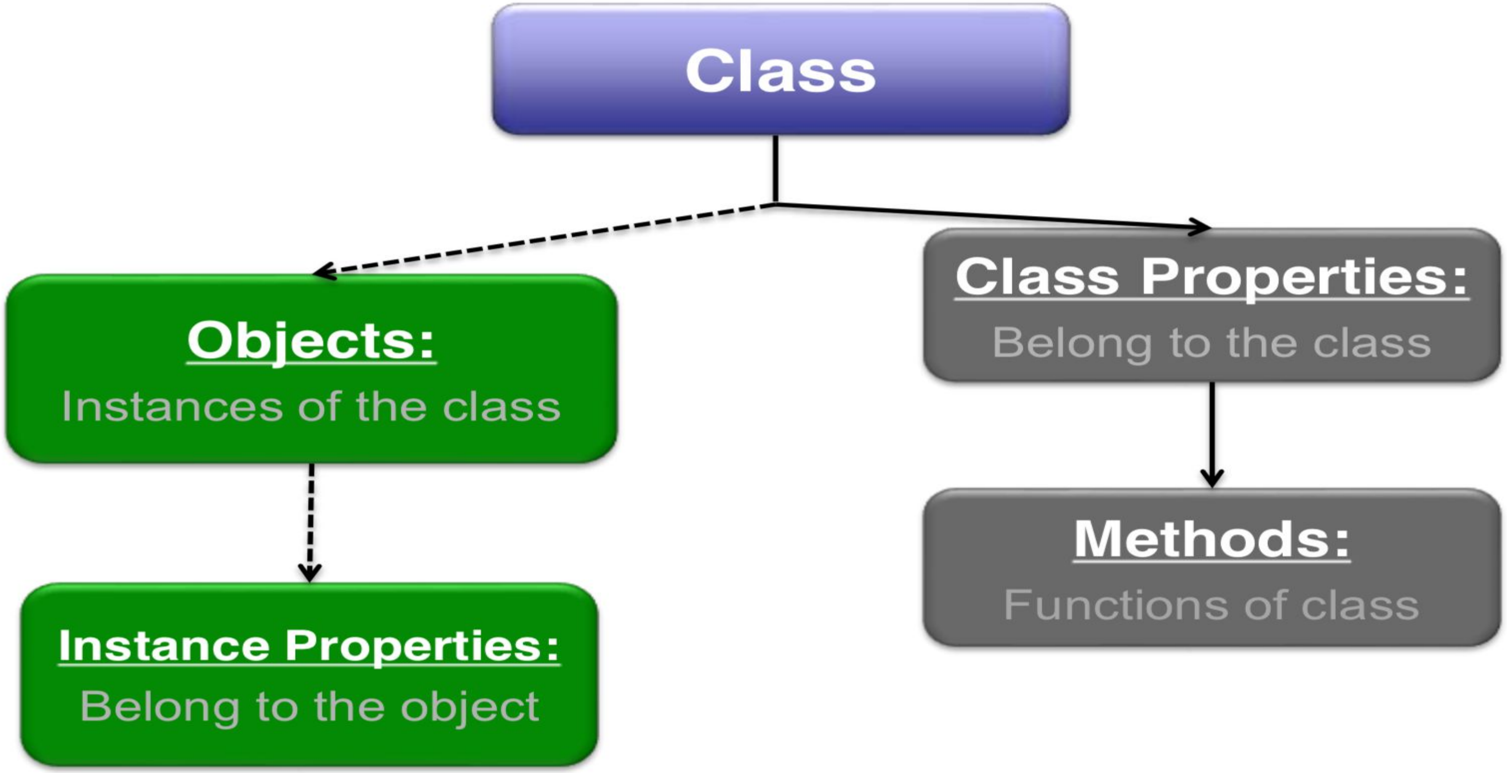
Car	Racing Car	Tank	Tricycle
Colour: *Red*	Colour: *Yellow*	Colour: *Green*	Colour: *Orange*
NumberOfDoors: *2*	NumberOfDoors: *0*	NumberOfDoors: *1*	NumberOfDoors: *0*
NumberOfWheels: *4*	NumberOfWheels: *4*	NumberOfWheels: *12*	NumberOfWheels: *3*
TopSpeed: *60mph*	TopSpeed: *200mph*	TopSpeed: *40mph*	TopSpeed: *50mph*

Object Oriented Paradigm


class
Time
hour
minute
void addMinutes(int m)
inTime
Attributes:
hour = 8
minute = 30
Methods:
void addMinutes(int m)
outTime
Attributes:
hour = 17
minute = 35
Methods:
void addMinutes(int m)
objects

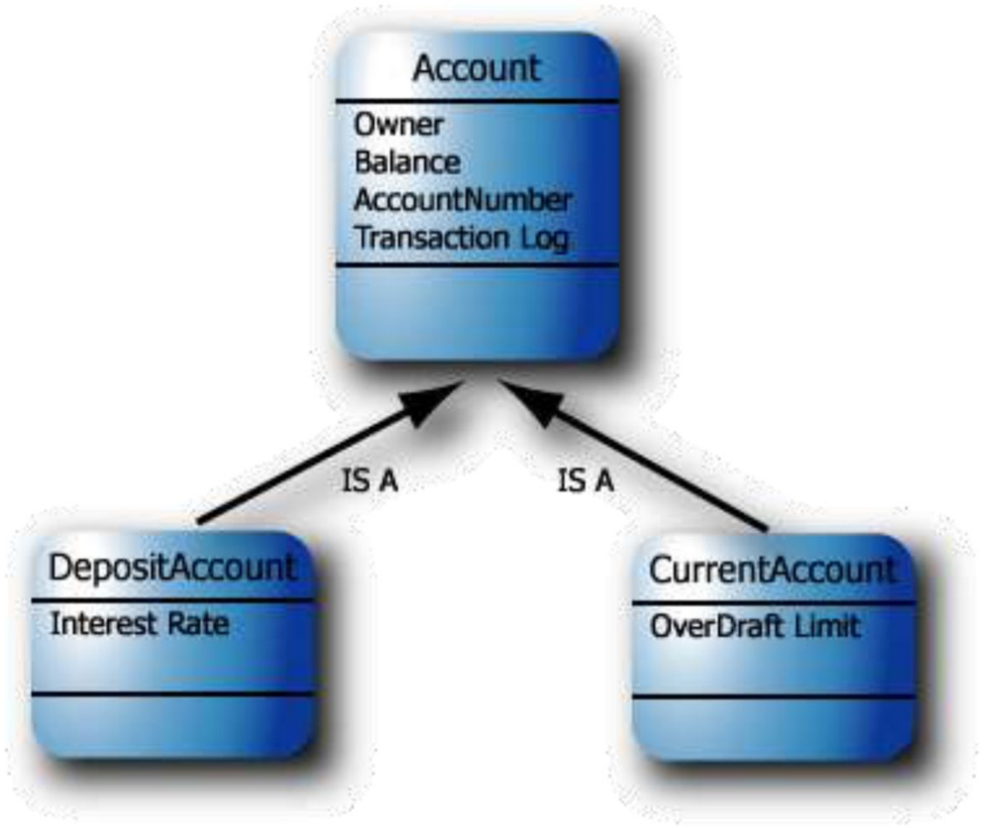
Classes *(objects)*

–



A class is a definition of objects with the same properties and the same methods.

Classes Example



What is an Object, again?

An object is an instance of a class

Each *copy of an object* from a particular class is called an *instance of the class.*

Instantiation

The act of creating a new instance of an class is called instantiation.

Inheritance=way of organizing classes

Term comes from inheritance of traits like eye color, hair color, and so on.

Classes with properties in common can be grouped so that their common properties are only defined once in parent class.

Inheritance=way of organizing classes

Superclass – inherit its attributes & methods to the subclass(es).

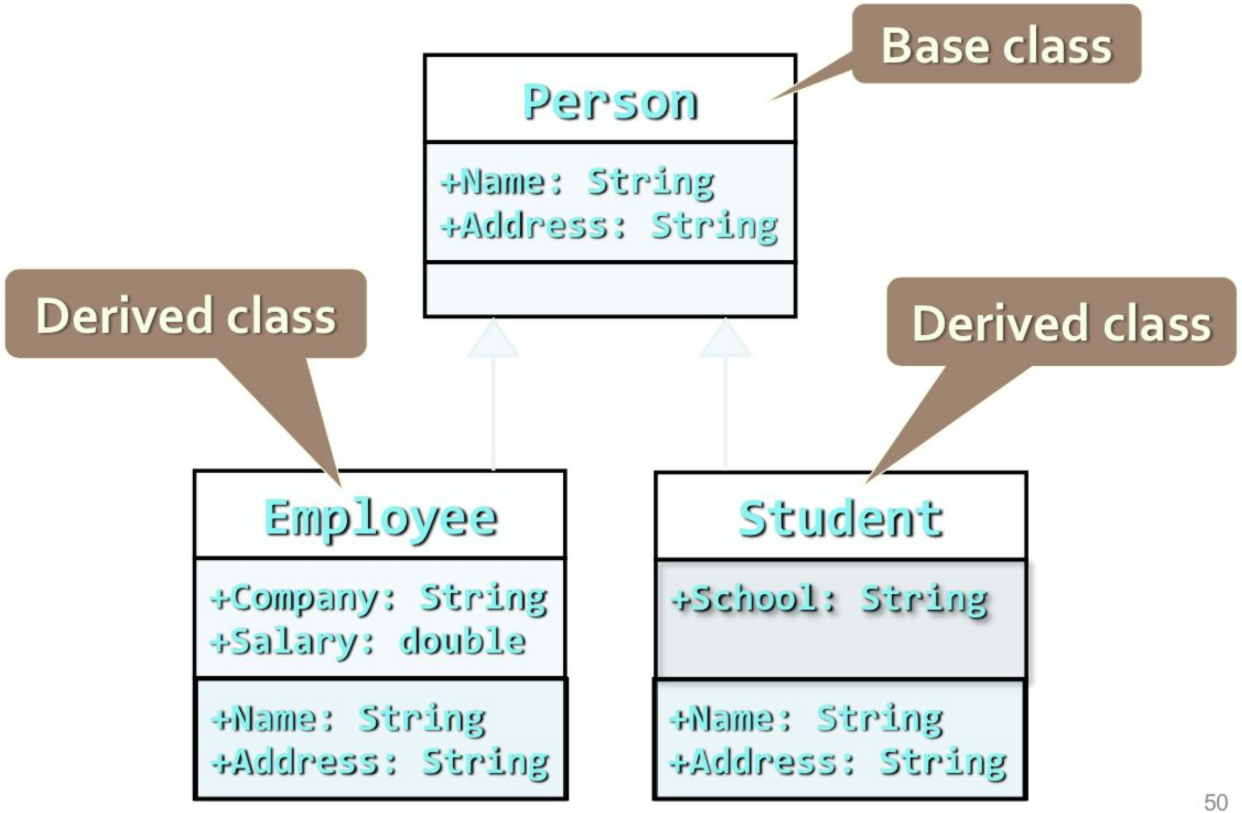
Subclass – can inherit all its superclass attributes & methods besides having its own unique attributes & methods.

Inheritance=way of organizing classes

Superclass – inherit its attributes & methods to the subclass(es).

Subclass – can inherit all its superclass attributes & methods besides having its own unique attributes & methods.

Inheritance – Example

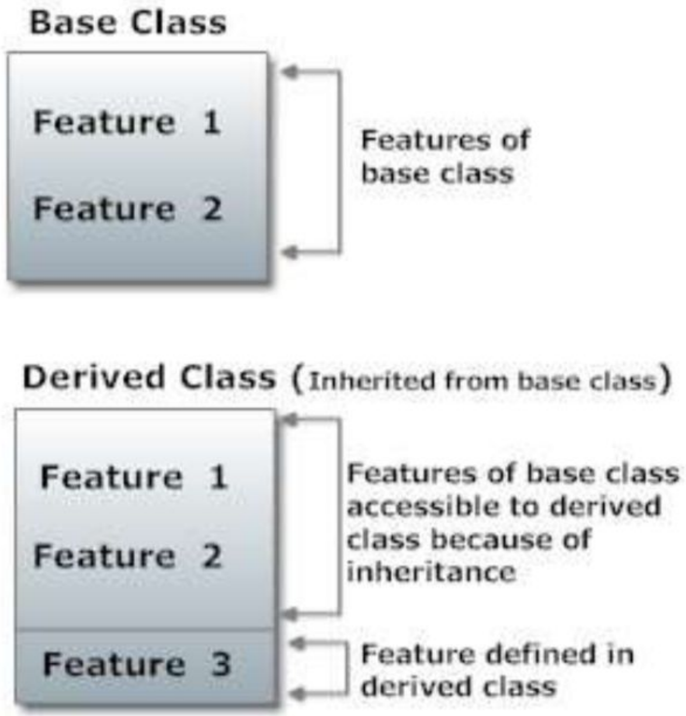


Inheritance - Use

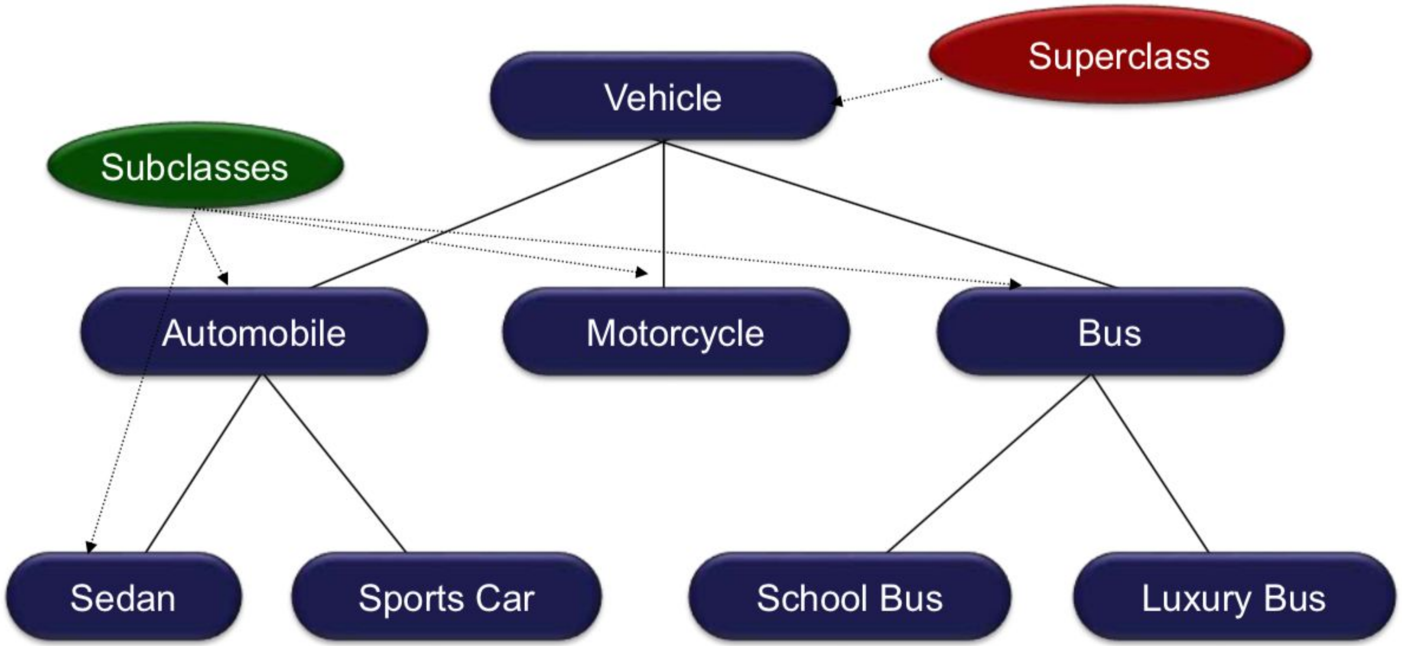
Helps in code organization

Allows code reusability

Inheritance

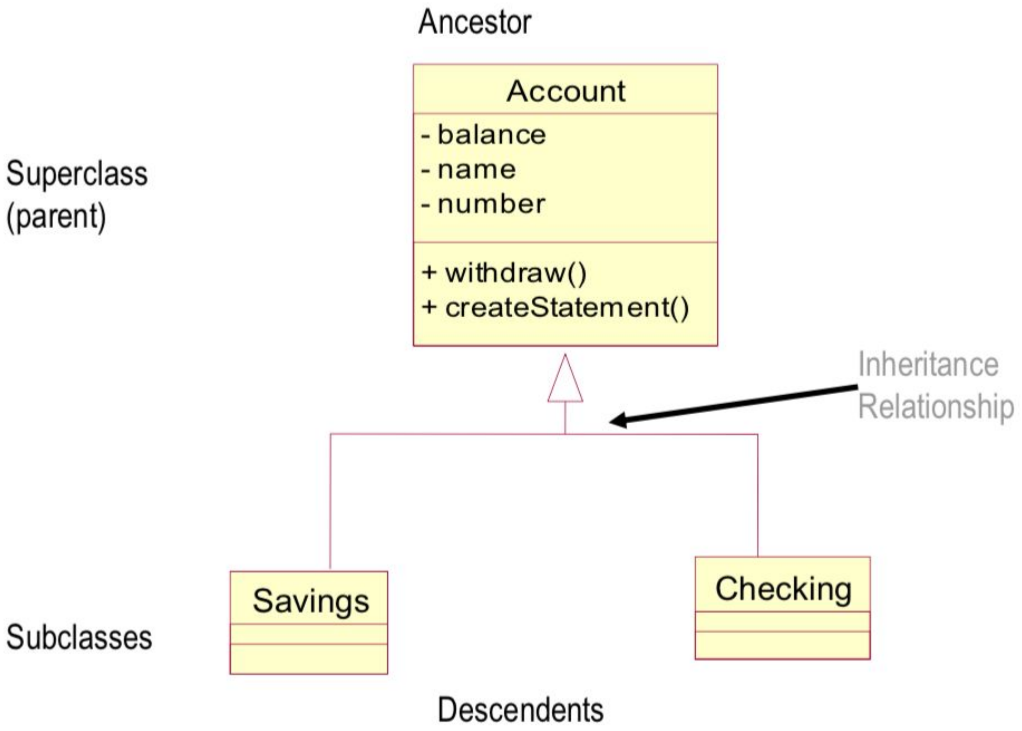

Base Class
Feature 1
Feature 2
Features of base class
Derived Class (Inherited from base class)
Feature 1
Feature 2
Features of base class accessible to derived class because of inheritance
Feature 3
Feature defined in derived class

An Inheritance Hierarchy



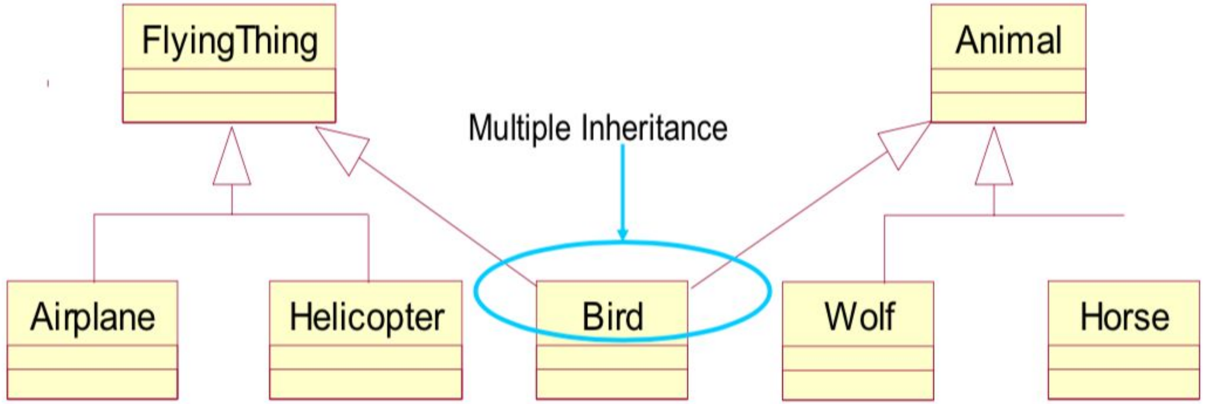
Example: Single Inheritance

One class inherits from another.



Example: Multiple Inheritance

- A class can inherit from several other classes.



Most modern languages don't support multiple inheritance!

Thank You