

INTRODUCTION TO OOPS

What is Object Oriented Programming?



- ❑ An object is like a black box.
- ❑ The internal details are hidden.

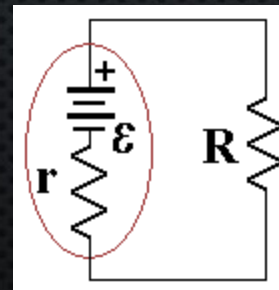
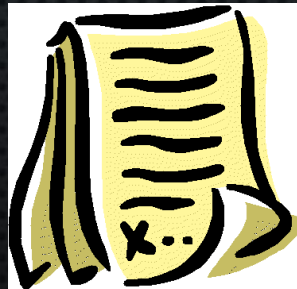
- ❑ Identifying *objects* and assigning *responsibilities* to these objects.
- ❑ Objects communicate to other objects by sending *messages*.
- ❑ Messages are received by the *methods* of an object

What is an object?

- ❑ Tangible Things as a car, printer, ...
- ❑ Roles as employee, boss, ...
- ❑ Incidents as flight, overflow, ...
- ❑ Interactions as contract, sale, ...
- ❑ Specifications as colour, shape, ...



ChandraSekhar(CS) Baratam



So, what are objects?

- ❑ An object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain.

Or

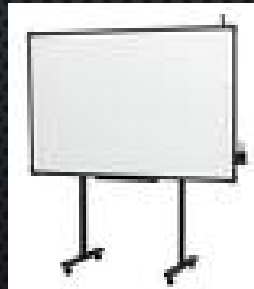
- ❑ An "object" is anything to which a concept applies.
Etc.

OBJECT



CLASS

Class : Class Room



Why do we care about objects?

- ❑ **Modularity** - large software projects can be split up in smaller pieces.
- ❑ **Reusability** - Programs can be assembled from pre-written software components.
- ❑ **Extensibility** - New software components can be written or developed from existing ones.

The two parts of an object

□ Object = Data + Methods

or to say the same differently:

□ An object has the responsibility to *know* and the responsibility to *do*.



ChandraSekhar(CS) Baratam

=



+



OOP



Abstraction

Encapsulation

Inheritance

Polymorphism

Basic Terminology

- ❑ **Encapsulation** is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.
- ❑ **Abstraction** is the representation of the essential features of an object. These are ‘encapsulated’ into an *abstract data type*.

Basic Terminology: Inheritance

❑ **Inheritance** means that one class inherits the characteristics of another class.

This is also called a “**is a**” relationship:

A car is a vehicle

A dog is an animal

A teacher is a person

Basic Terminology: Polymorphism

- ❑ **Polymorphism** means “having many forms”. It allows different objects to respond to the same message in different ways, the response specific to the type of the object.
- ❑ Example: `move()`



Basic Terminology: Aggregation

- Aggregation describes a “has *a*” relationship. One object is a part of another object.

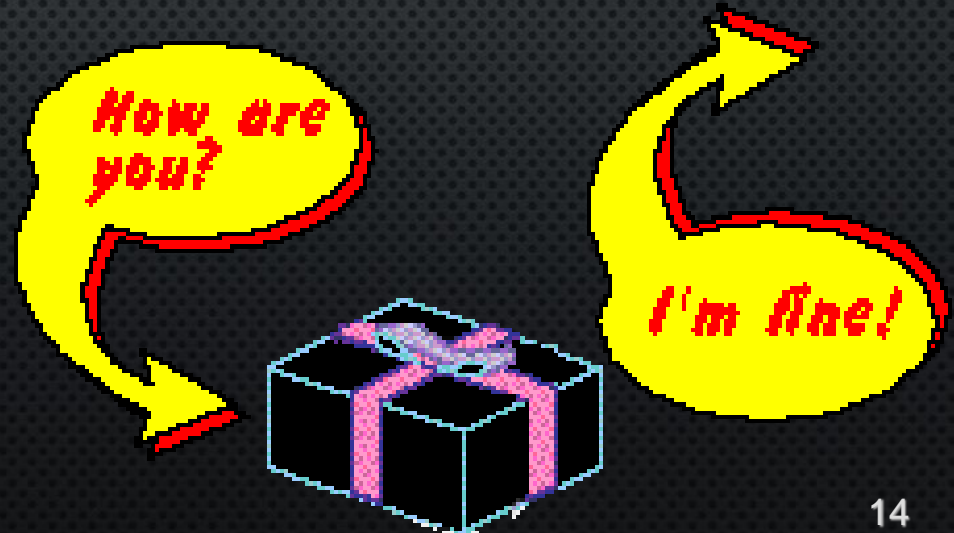
A car has wheels.

- We distinguish between *composite* aggregation (the composite “owns” the part) and *shared* aggregation (the part is shared by more than one composite).

Basic Terminology:

Behaviour and Messages

- The most important aspect of an object is its *behaviour* (the things it can do). A behaviour is initiated by sending a *message* to the object (usually by calling a method).



The two steps of Object Oriented Programming

- ❑ **Making Classes:** Creating, extending or reusing abstract data types.
- ❑ **Making Objects interact:** Creating objects from abstract data types and defining their relationships.