SOOAD

UNIT 3

OBJECT ORIENTED ANALYSIS & DESIGN

UNIT -3 OOAD

- Introduction
- Object-Oriented Modelling
- Object-Oriented Approach
- The Constituents of OOAD
- Pillars of Object-Oriented Analysis and Design
- The Language of OOAD – Unified Modelling Language

Introduction

- Object-oriented analysis and design (OOAD) is a technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modeling throughout the software development process to guide stakeholder communication and product quality.
- OOAD in modern software engineering is typically conducted in an iterative and incremental way.
- The outputs of OOAD activities are analysis models (for OOA) and design models (for OOD) respectively.
- The intention is for these to be continuously refined and evolved, driven by key factors like risks and business value.

Introduction

- The first object–oriented language was Simula (Simulation of real systems) that was developed in 1960 by researchers at the Norwegian Computing Center.
- In 1970, Alan Kay and his research group at Xerox PARK created a personal computer named Dynabook and the first pure object-oriented programming language (OOPL) - Smalltalk, for programming the Dynabook.
- In the 1980s, Grady Booch published a paper titled Object Oriented Design that mainly presented a design for the programming language, Ada. In the ensuing editions, he extended his ideas to a complete object–oriented design method.
- In the 1990s, Coad incorporated behavioral ideas to object-oriented methods.
- The other significant innovations were Object Modeling Techniques (OMT) by James Rum Baugh and Object-Oriented Software Engineering (OOSE) by Ivar Jacobson.

OBJECT ORIENTED MODELLING

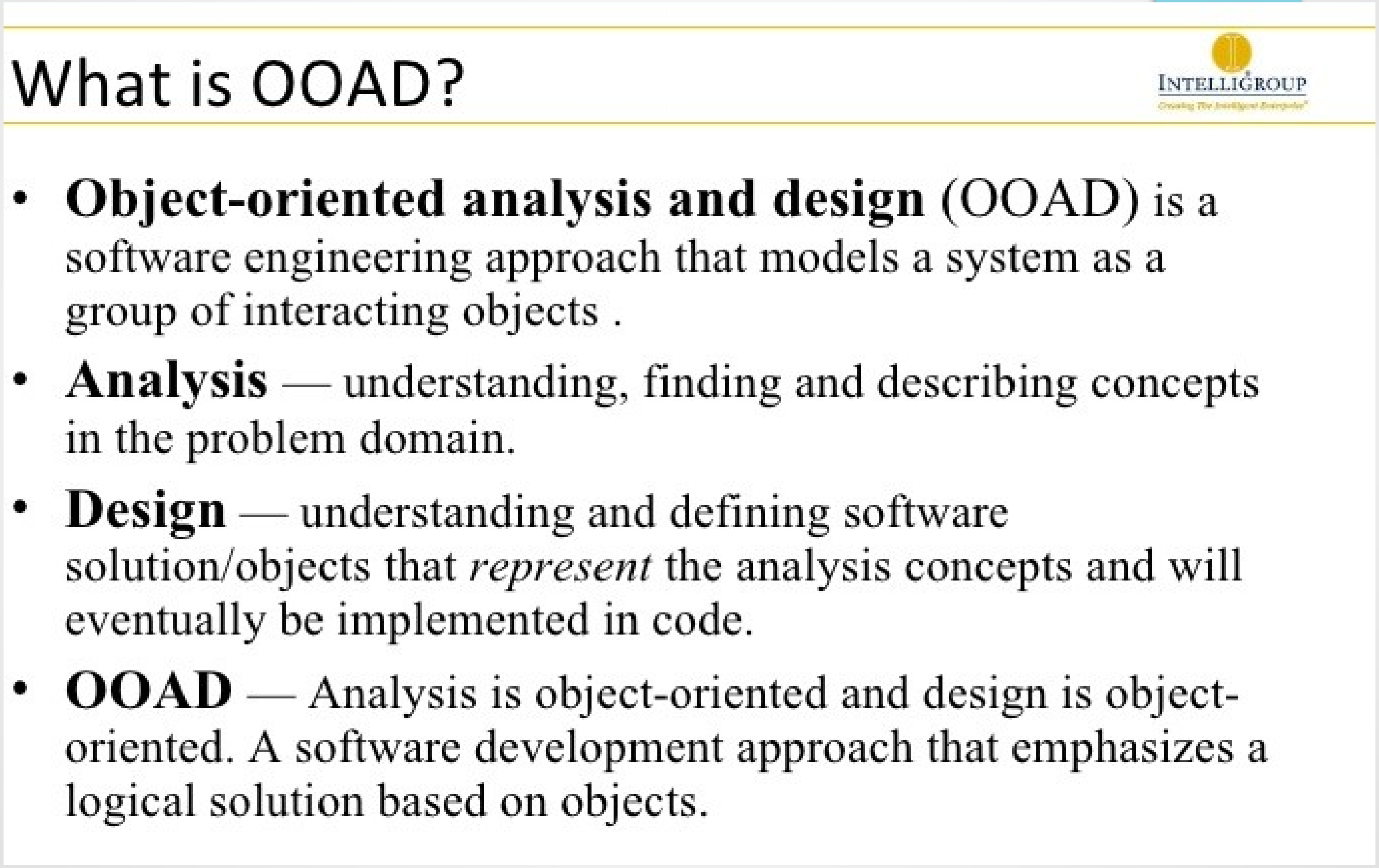
Object Oriented Modelling

- Object Oriented Analysis & Design (OOAD) encompasses on
  - Objects
  - Analysis
  - Design

- OOAD is a software engineering approach that models a system as group of interacting objects.

- Each object represents some entity of interest in the system being modelled and characterized by its
  - class
  - state
  - behaiviour

OOAD



What is OOAD?

INTELLIGROUP

- **Object-oriented analysis and design** (OOAD) is a software engineering approach that models a system as a group of interacting objects .
- **Analysis** — understanding, finding and describing concepts in the problem domain.
- **Design** — understanding and defining software solution/objects that *represent* the analysis concepts and will eventually be implemented in code.
- **OOAD** — Analysis is object-oriented and design is object-oriented. A software development approach that emphasizes a logical solution based on objects.

Object Oriented Modelling

- Various models are used to show the static structure, dynamic behaviour and run-time deployment of these collaborating objects.
- Different Models for different phases are:
  - Analysis Model
  - Architecture Model
  - Component Model

Object Oriented Modelling

- Analysis Model:
  - Model of existing system
  - The user’s requirement
  - a high-level understanding of a possible solution to those requirements.

Object Oriented Modelling

- Architecture Model:
  - Evolving model
  - structure of the solution to the requirements defined in analysis model
  - Primary focus is on architecture:
    - Components
    - interfaces and
    - the structure of the solution,
    - the deployment of structure across nodes and
    - trade-offs and decisions that lead up to that structure

Object Oriented Modelling

- Component (Design) Model:
  - Number of models
  - one per component shows internal structure of the pieces of the architecture model.
  - Detailed class structure of its component
  - Attributes, operations, dependencies, and the behaviour of its classes.

Why to do Modelling?

- Can make model changes in development model
- To catch costly bugs early
- Eary detection and correction can save a lot on the cost and schedule of a bug fix.

OOA vs. OOD

- Object Oriented Analysis (OOA) applies object-modelling techniques to analyze the functional requirements for a system
- Object Oriented Design (OOD) elaborates the analysis models to produce implementation specifications.

OOA focuses on what the system does whereas

OOD focuses on how the system does it.

OBJECT ORIENTED APPROACH

Object Oriented Approach

First step of OO methodology is concerned with understanding the domain and modelling the real world application for problem statement formulation.

OOAD Stages:

- Analysis stage
- Design stage
- Implementation stage

Object Oriented Approach

- Analysis stage - produces SRS (Software/ System Requirement Specification)
  - Consists of:
    - Business process diagrams
    - Use-case diagrams
    - Class and object diagram

Object Oriented Approach

- Design Phase – Database Design is arrived from analysis stage.
  - Involves
    - Sequence diagram
    - Collaboration diagram
    - Activity diaram
    - State-chart diaram
- Implementation Phase – clear modular structure for programs where implementation details are hidden
  - Produces
    - Component diagram
    - Deployment diagram

Object Oriented Programming

- OOP is primarily concerned with programming language and software implementation issues.
- OOP makes it easy to maintain and modify existing code
- OOP provides a good framework for code libraries
- Code can be easily adopted and modified by a programmer.

Object Orientation

- Object Oriented (OO) means organize software as a collection of discrete objects.
- It has both data structure and behaviour.
- Map natuarally to real-world objects.
- Supports abstraction at object level.
- Development can proceed at the object level
- Designing, coding, testing and maintaining the system much simpler.
- Moving from one phase to another does not require different styles and methodologies
- Reduce complexity and makes clear system development
- Nice syntatic mechanism for achieving some classic aspects of well-designed code.

Object Orientation Analysis

- Examines problem domain
- Producing a conceptual model of the information that exists in the area being examined.
- Do not consider implementation problems.
- Analysis is done before design
- Sources of analysis:
  - written statement
  - formal document
  - Interviews with stakeholders

Object Orientation Analysis

- System is divided into multiple domains, representing different business, technological and analyzed seperately.

Result of OOA is a description of what the system is functionally required to do, in the form of a conceptual model.

A set of USE-CASES, one or more UML class diagrams, and a number of interaction diagrams.

OOA is the process of defining the problems in terms of real world objects.

Object Orientation Analysis

- OOA is the process of defining the problem in terms of objects:
  - real world objects with which system must interact, and candidate software objects used to explore various solutions aalternatives.
- One can define all of the real-world objects in terms of their classes, attributes and operations.

Object Orientation Design

- OOA means defining the problem and OOD is the process of defining the solution.
- Defining the ways in which the system is prepared as per analysis phase.
- "Object Oriented Design (OOD) is the process of defining the components, interfaces, objects, classes, attributes and operations that satisfies funtional requirements."
- Start with candidate objects defined during analysis, but can add much more rigour to their definitions.
- Then we add or change objects as needed to refine a solution.

Object Orientation Design

- Two Scales of Design:
  - *Achitectural design* - defining components of system
  - *Component design* – defining classes and interfaces
  - OOD transforms the conceptual model to take account of constraints imposed by the chosen architecture and any non functional constraints, like transaction throughput, response time, run-time platform, developement environment, or programming language.
  - The concepts in the analysis model are mapped onto implementation classes and interfaces.
  - The result is a model of the solution domain, a detailed description of how the system is to be built.

THE CONSTITUENTS OF OOAD

The Constituents Of OOAD

- Objects
- Classes
- Links
- Association
- Generalization
- Specialization
- Aggregation
- Composition

Object

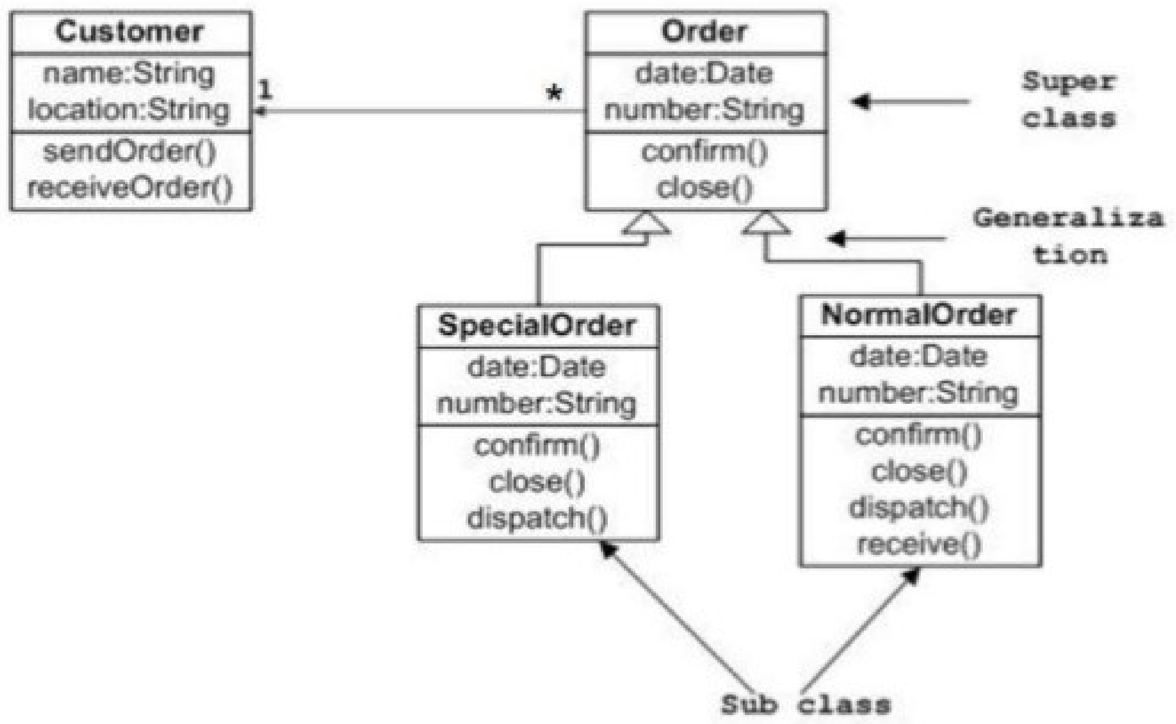
- An object is the foundation of an OOP.
- Basic run-time entities in an OO system.
- Problems are identified in terms of objects.
- Object interact with each other by sending messages.
- Without knowing the details of their data or code.
- Objects improves program reliability, simplify software maintenance and management of libraries.
- Object is identified by its identity that distingushes it from other objects and its behaviour.

Class

- Class is a blue print or factory that defines the abstract characteristics of an object including its attributes, fields or properties and its abstract behaviour likes its methods, operations or features.
- Collection of objects of similar types.
- Attributes: state that describes an object.
- Operations (Methods): is a behaviour that an object can perform.

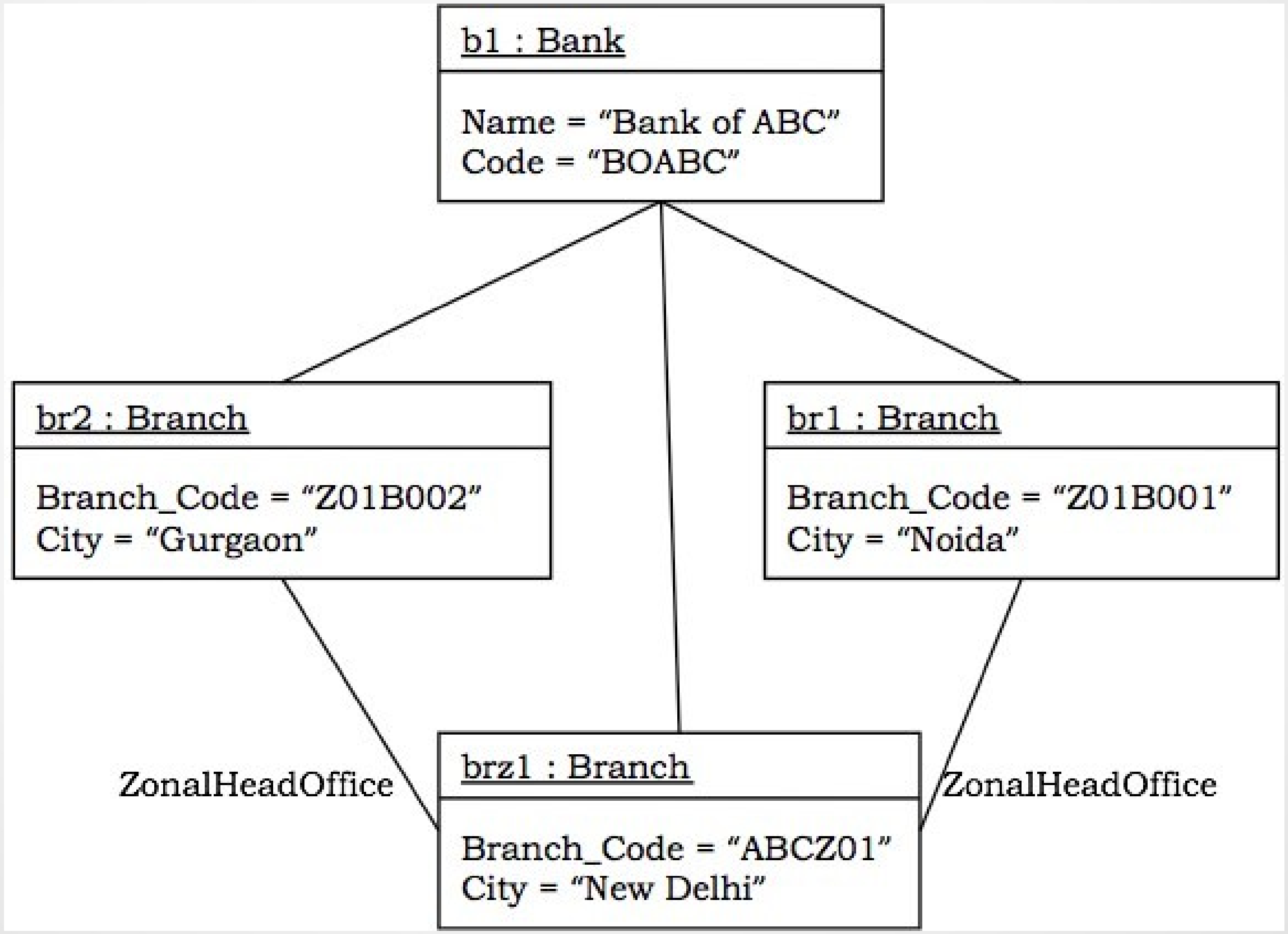
Class

Class Diagrams



A sample class diagram

Class


b1 : Bank
Name = "Bank of ABC"
Code = "BOABC"
br2 : Branch
Branch_Code = "Z01B002"
City = "Gurgaon"
br1 : Branch
Branch_Code = "Z01B001"
City = "Noida"
brz1 : Branch
Branch_Code = "ABCZ01"
City = "New Delhi"
ZonalHeadOffice
ZonalHeadOffice

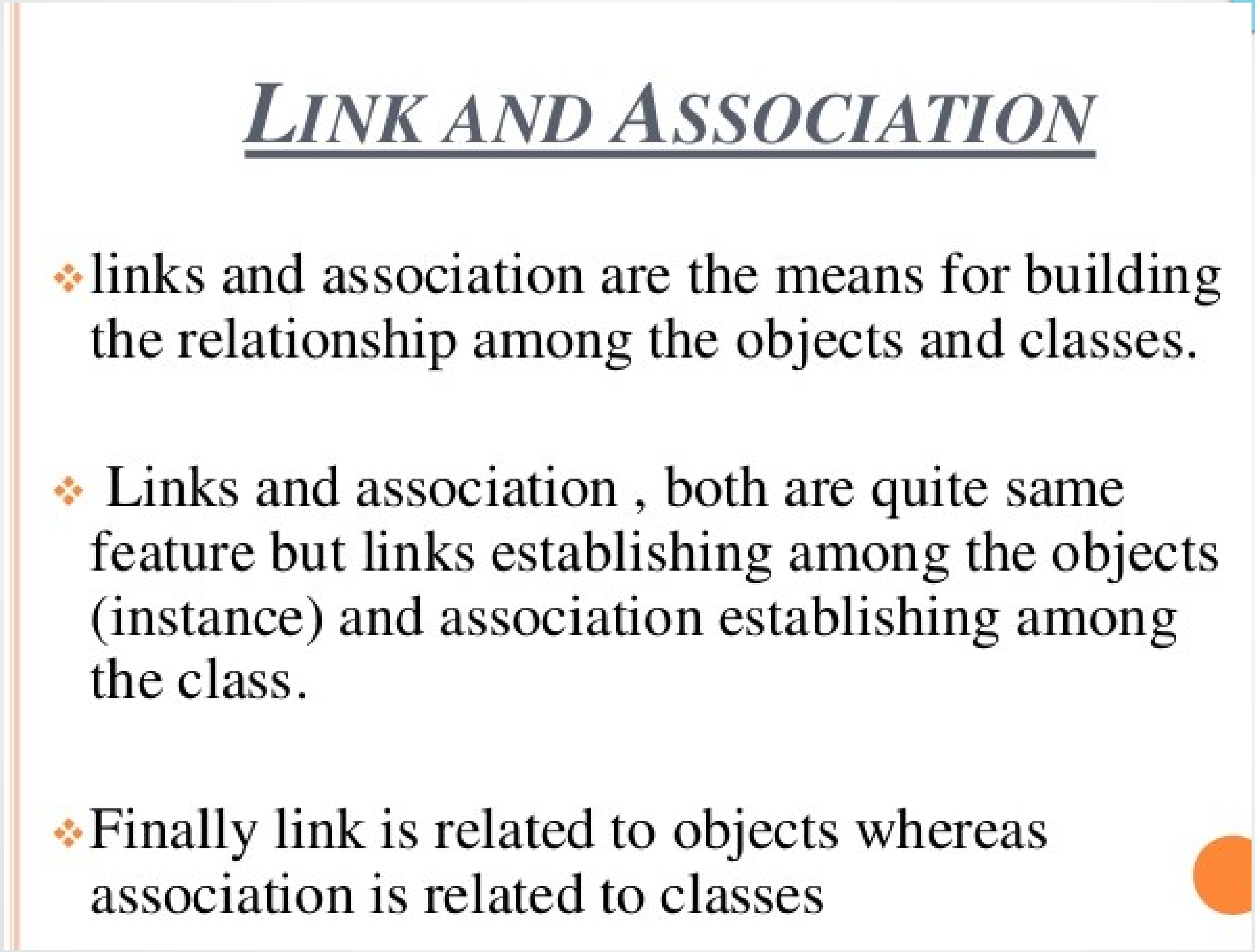
Links

- A link is a relationship among instance of classes (objects).
- To represent relationship between two objects.
- A link represents a connection through which an object collaborates with other objects.
- It as “a physical or conceptual connection between objects”.
- A link depicts the relationship between two or more objects.

Association

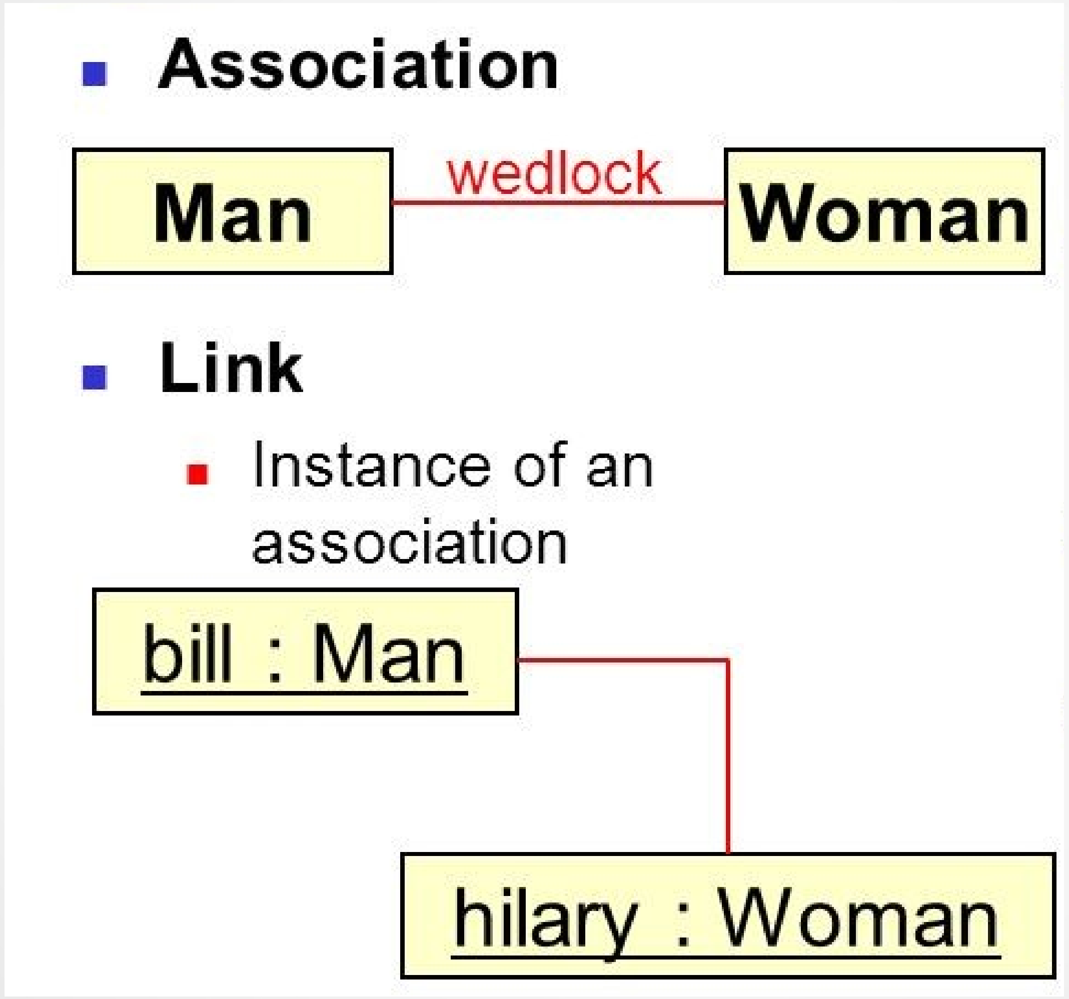
- Association is used to represent the relationship between the two classes.
- Example: A Student and a Faculty are having an association.
- Association is a group of links having common structure and common behavior.
- Association depicts the relationship between objects of one or more classes.
- A link can be defined as an instance of an association.

Links & Association


LINK AND ASSOCIATION

- links and association are the means for building the relationship among the objects and classes.
- Links and association , both are quite same feature but links establishing among the objects (instance) and association establishing among the class.
- Finally link is related to objects whereas association is related to classes

Links & Association


- **Association**

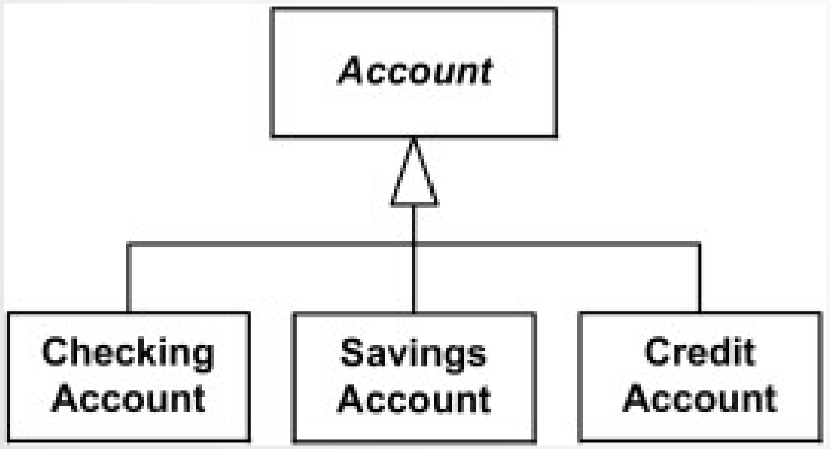
- **Link**
  - Instance of an association

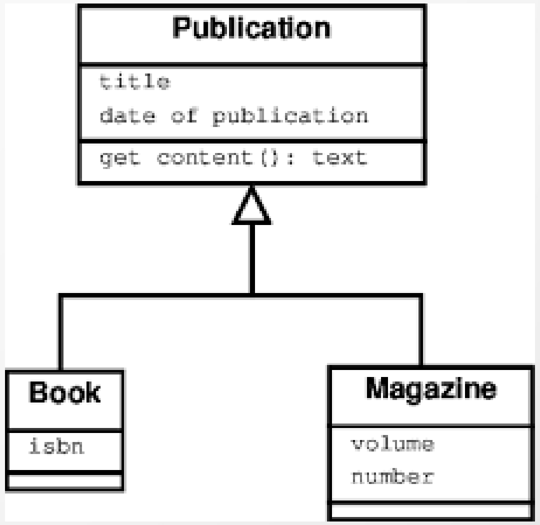
Generalization

- Act of identifying and categorizing similar objects into classes.
- In the generalization process, the common characteristics of classes are combined to form a class in a higher level of hierarchy, i.e., subclasses are combined to form a generalized super-class.
- It represents an "is – a – kind – of" relationship.
- For example, "car is a kind of land vehicle", or "ship is a kind of water vehicle".

Generalization

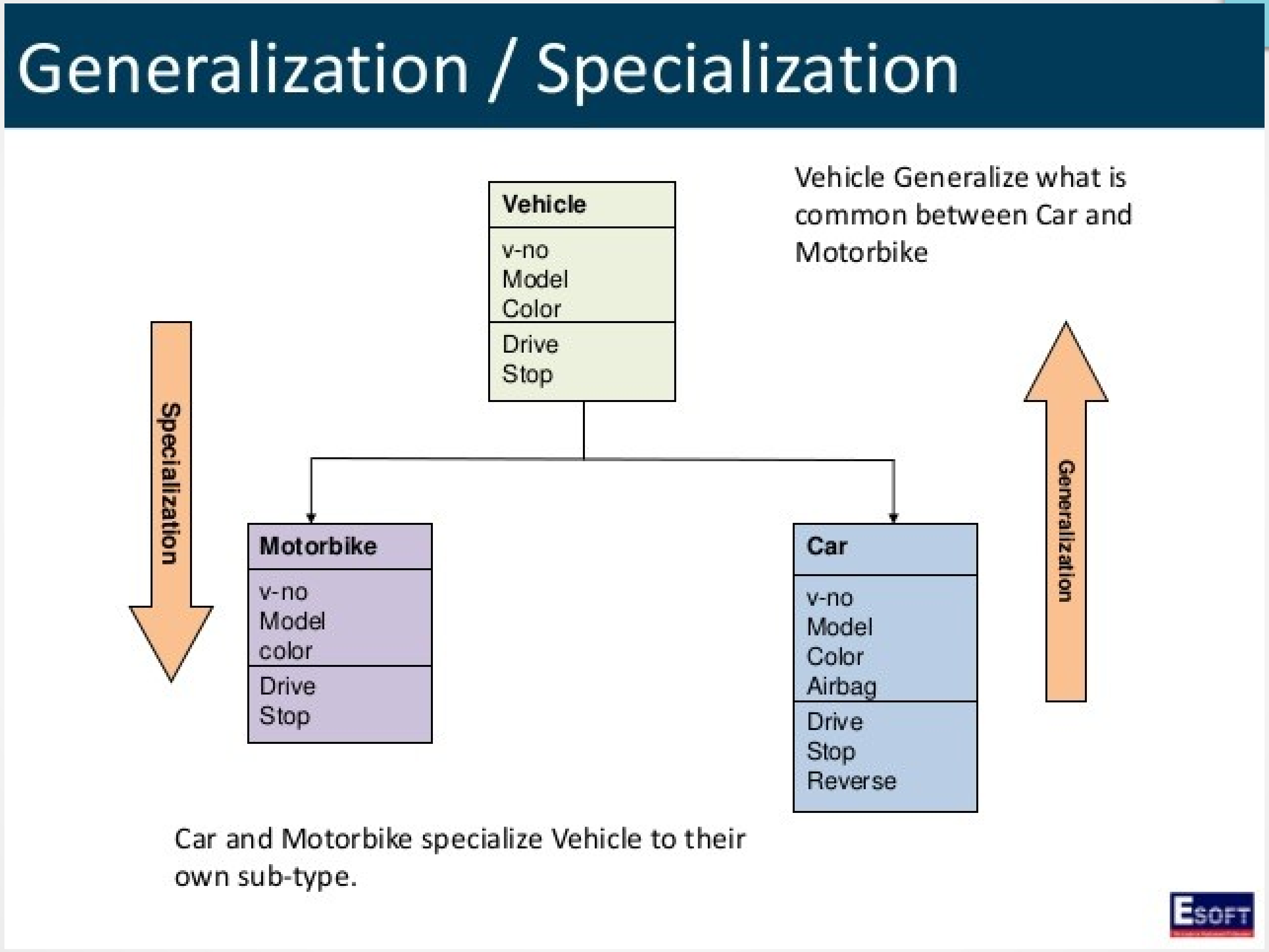
- Generalization is the process of extracting shared characteristics from two or more classes, and combining them into a generalized superclass.
- Shared characteristics can be attributes, associations, or methods.





Specialization

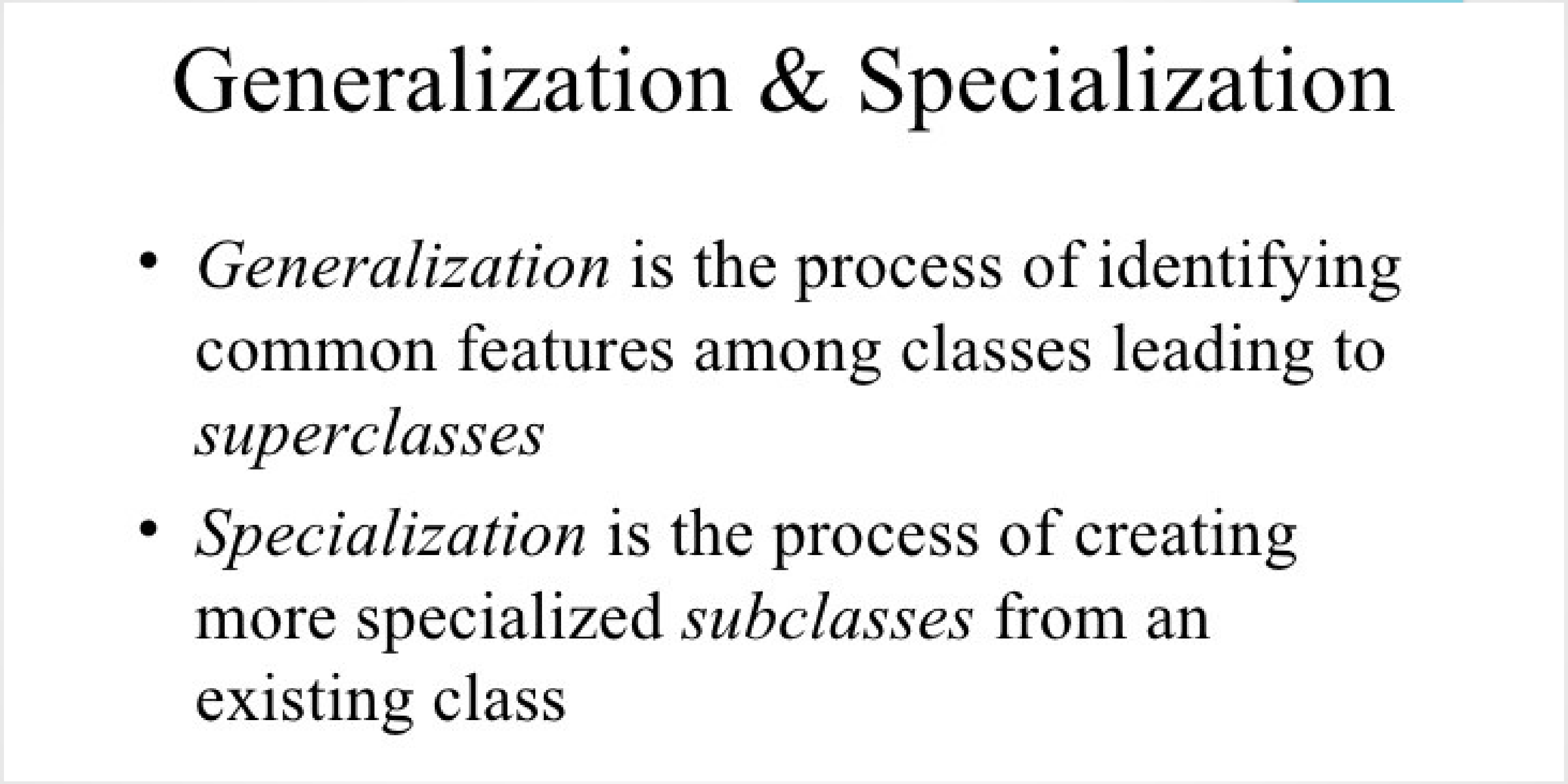
- In contrast to generalization, specialization means creating new subclasses from an existing class.
- If it turns out that certain attributes, associations, or methods only apply to some of the objects of the class, a subclass can be created.
- Here, the distinguishing features of groups of objects are used to form specialized classes from existing classes.
- It can be said that the subclasses are the specialized versions of the super-class.



Generalization / Specialization

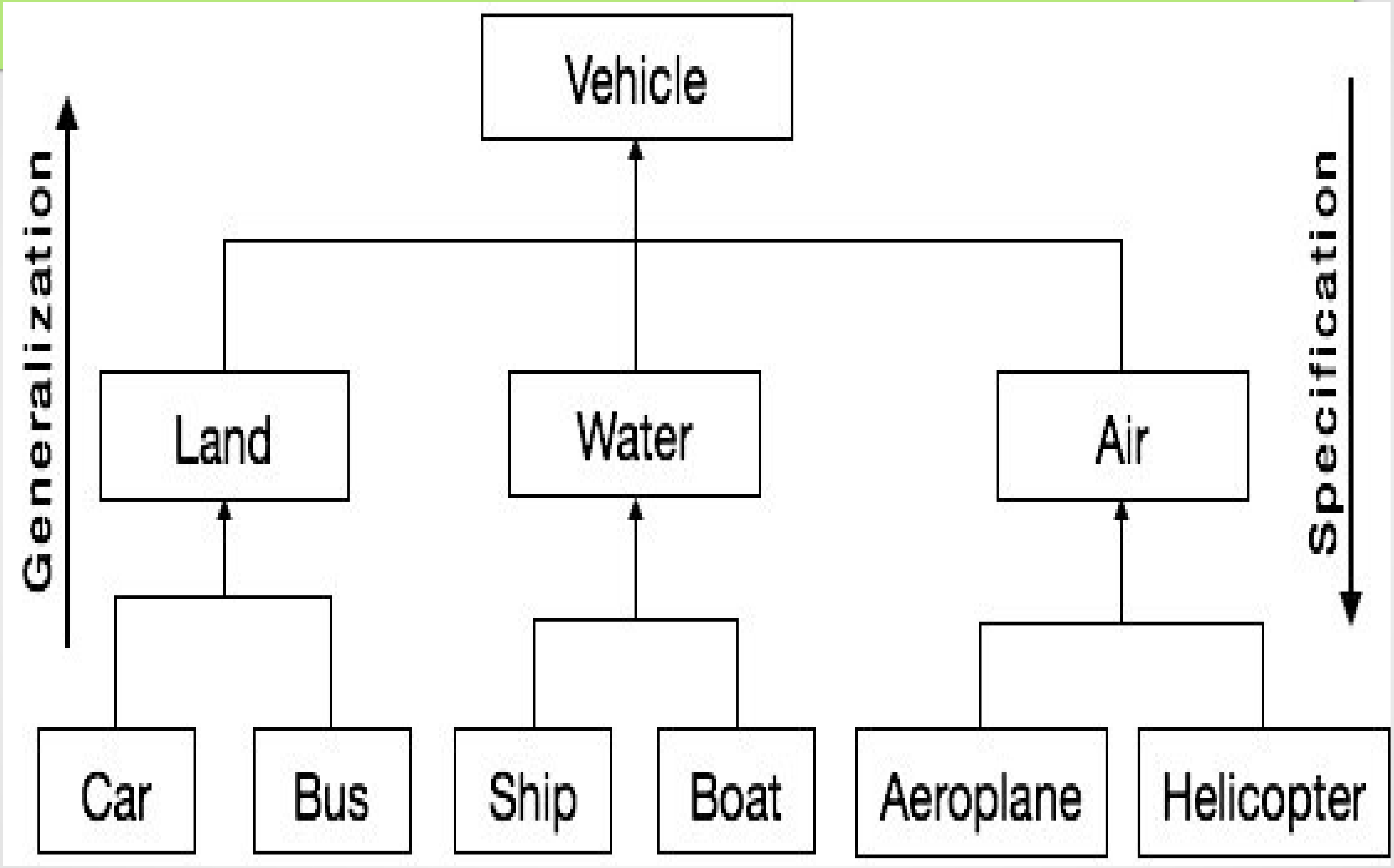
Vehicle Generalize what is common between Car and Motorbike

Car and Motorbike specialize Vehicle to their own sub-type.



Generalization & Specialization

- *Generalization* is the process of identifying common features among classes leading to *superclasses*
- *Specialization* is the process of creating more specialized *subclasses* from an existing class

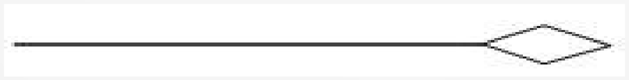

Vehicle
Land
Water
Air
Car
Bus
Ship
Boat
Aeroplane
Helicopter
Generalization
Specification

Aggregation or Composition

- Aggregation or composition is a relationship among classes by which a class can be made up of any combination of objects of other classes.
- It allows objects to be placed directly within the body of other classes.
- Aggregation is referred as a "part–of" or "has–a" relationship, with the ability to navigate from the whole to its parts.
- An aggregate object is an object that is composed of one or more other objects.

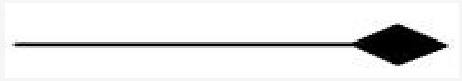
Aggregation or Composition

- Aggregation is a special case of association.
- A directional association between objects.
- When an object 'has-a' another object, then you have got an aggregation between them.
- Direction between them specified which object contains the other object.
- Aggregation is also called a "Has-a" relationship.



Composition

- Composition is a special case of aggregation.
- In a more specific manner, a restricted aggregation is called composition.
- When an object contains the other object, if the contained object cannot exist without the existence of container object, then it is called composition.
- Example: A class contains students. A student cannot exist without a class. There exists composition between class and studen



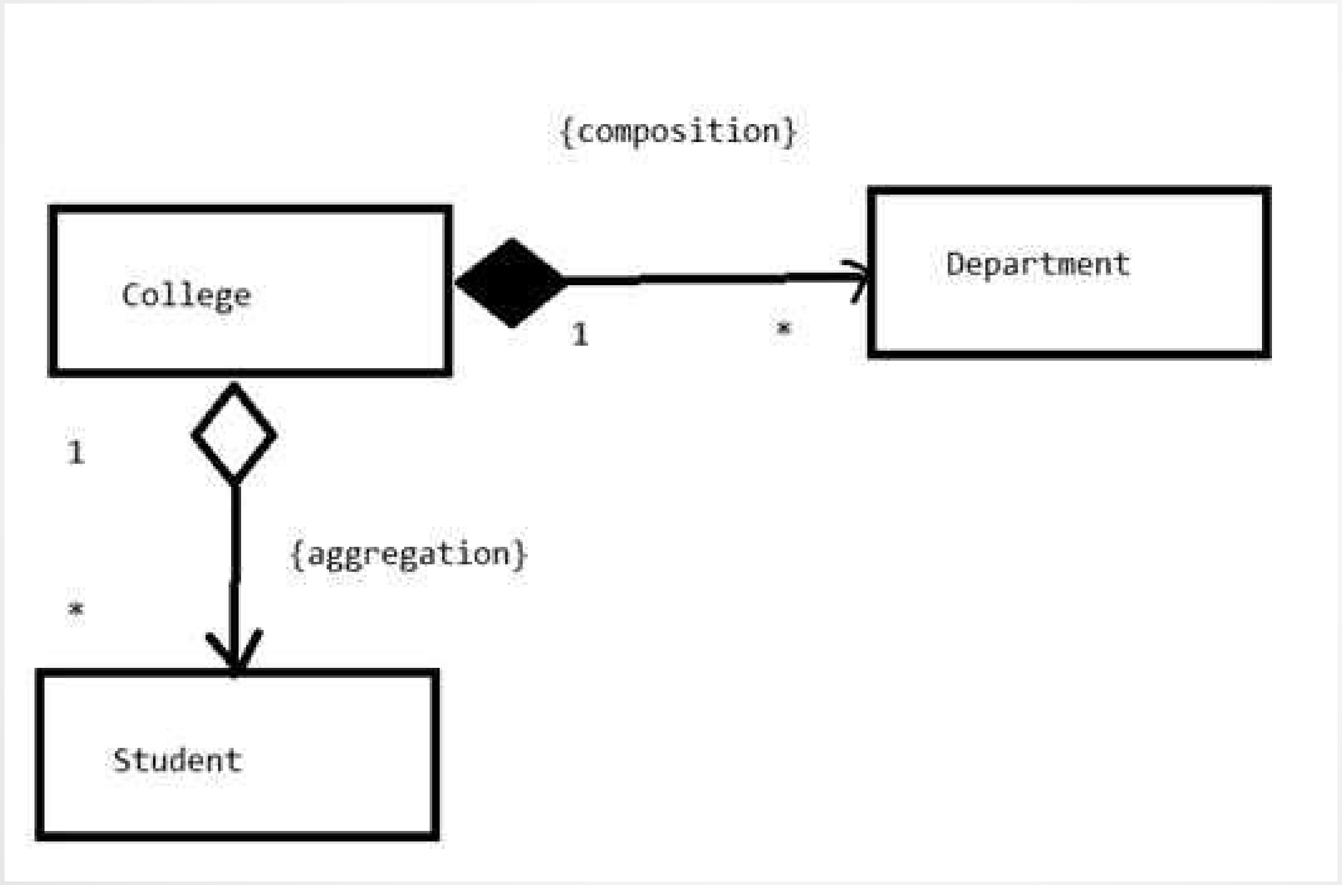
Difference between aggregation and composition

- Composition is more restrictive. When there is a composition between two objects, the composed object cannot exist without the other object. This restriction is not there in aggregation.
- Though one object can contain the other object, there is no condition that the composed object must exist.
- The existence of the composed object is entirely optional.
- In both aggregation and composition, direction is must. The direction specifies, which object contains the other object.

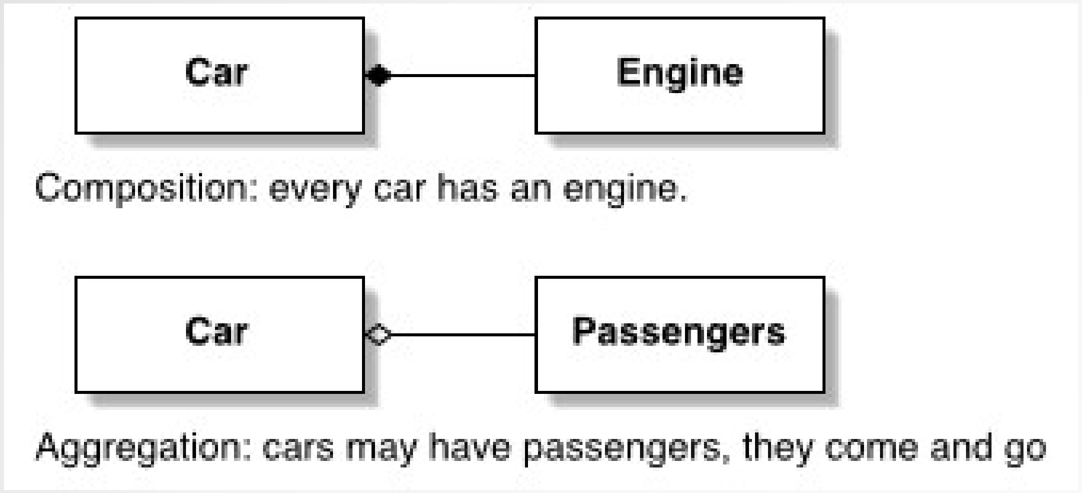
Difference between aggregation and composition

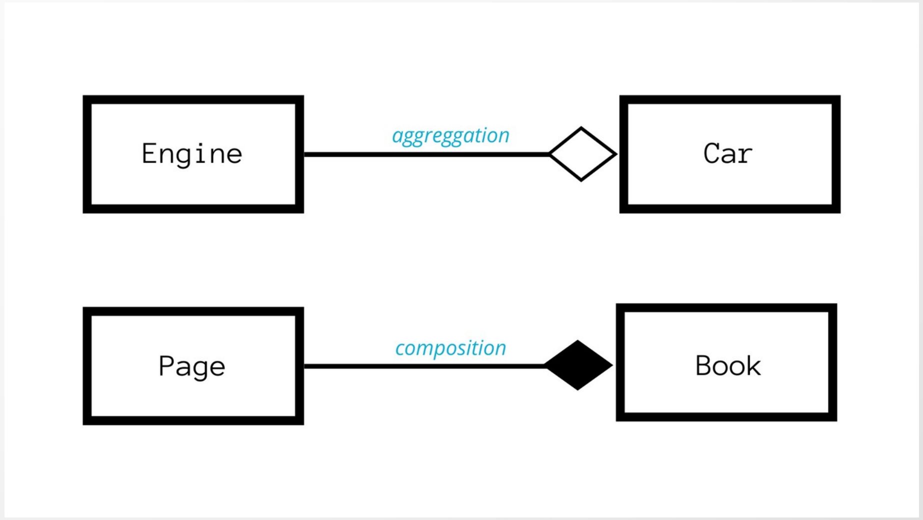
- Example:
  - A Library contains students and books.
- Relationship between library and student is aggregation.
- Relationship between library and book is composition.
- A student can exist without a library and therefore it is aggregation.
- A book cannot exist without a library and therefore its a composition.

Difference between aggregation and composition



Difference between aggregation and composition


Car
Engine
Composition: every car has an engine.
Car
Passengers
Aggregation: cars may have passengers, they come and go


Engine
aggreggation
Car
Page
composition
Book