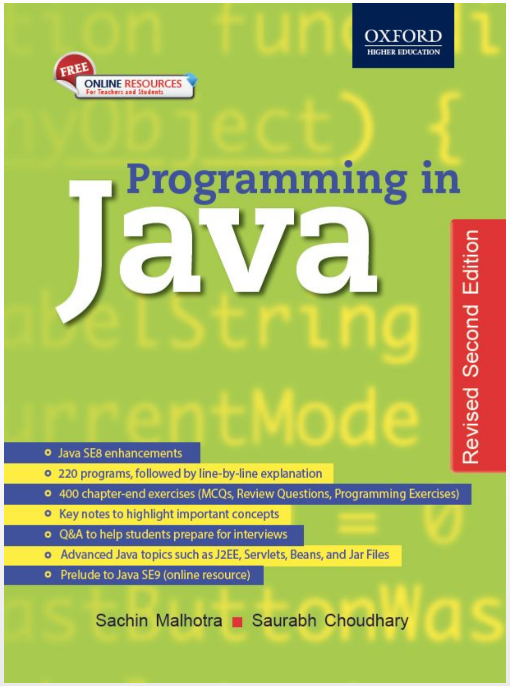
OXFORD
UNIVERSITY PRESS



OXFORD
HIGHER EDUCATION

FREE ONLINE RESOURCES For Teachers and Students

Programming in Java

Revised Second Edition

- Java SE8 enhancements
- 220 programs, followed by line-by-line explanation
- 400 chapter-end exercises (MCQs, Review Questions, Programming Exercises)
- Key notes to highlight important concepts
- Q&A to help students prepare for interviews
- Advanced Java topics such as J2EE, Servlets, Beans, and Jar Files
- Prelude to Java SE9 (online resource)

Sachin Malhotra ■ Saurabh Choudhary

Programming in Java

Revised 2nd Edition

Sachin Malhotra & Saurabh Choudhary

Chapter 1

Introduction to OOP

Objective

- Understand what is Object Oriented programming
- Understand the principles of OOP
- How is OOP different from Procedural languages
- Problems in Procedural programming and how OOP overcomes them
- Applications of OOP
- Understand UML notations

Procedural Languages

- C, PASCAL, FORTRAN languages are all procedural languages.
- A program is a list of instructions.
- Complexity increases as the length of a program increases.
- Divide a large program into different functions or modules.
- Problems with Procedural languages.
    - functions have unrestricted access to global data
    - they provide poor mapping to real world
    - Not extensible

Introduction to OOP

- A programming paradigm.
- Deals with concepts of object to build programs and software applications.
- Modeled around real world.
- The world we live in is full of objects.
- Every object has a well-defined identity, attributes and behavior.
- Objects exhibit the same behavior in programming.

OOP Principles

- Classes
- Objects
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Class

- Blueprint for an object, a plan, or template.
- Description of a number of similar objects is also called class.
- A class is also defined as a new data type; a user-defined type.
- Defining a class doesn’t create an object.
- Classes are logical in nature.
- For examples, Furniture do not have any existence but tables and chairs do exist.

Object

- Defined as instance of a class. For example table, chair are all instances of the class Furniture.
- Objects have unique identity, state and behavior.
- State is defined by the attributes of the object.
- Different objects have different attributes (characteristics)
  - For example the attributes of student are name, roll number etc.
- Behavior actually determines the way an object interacts with other objects.
- Synonym to functions.

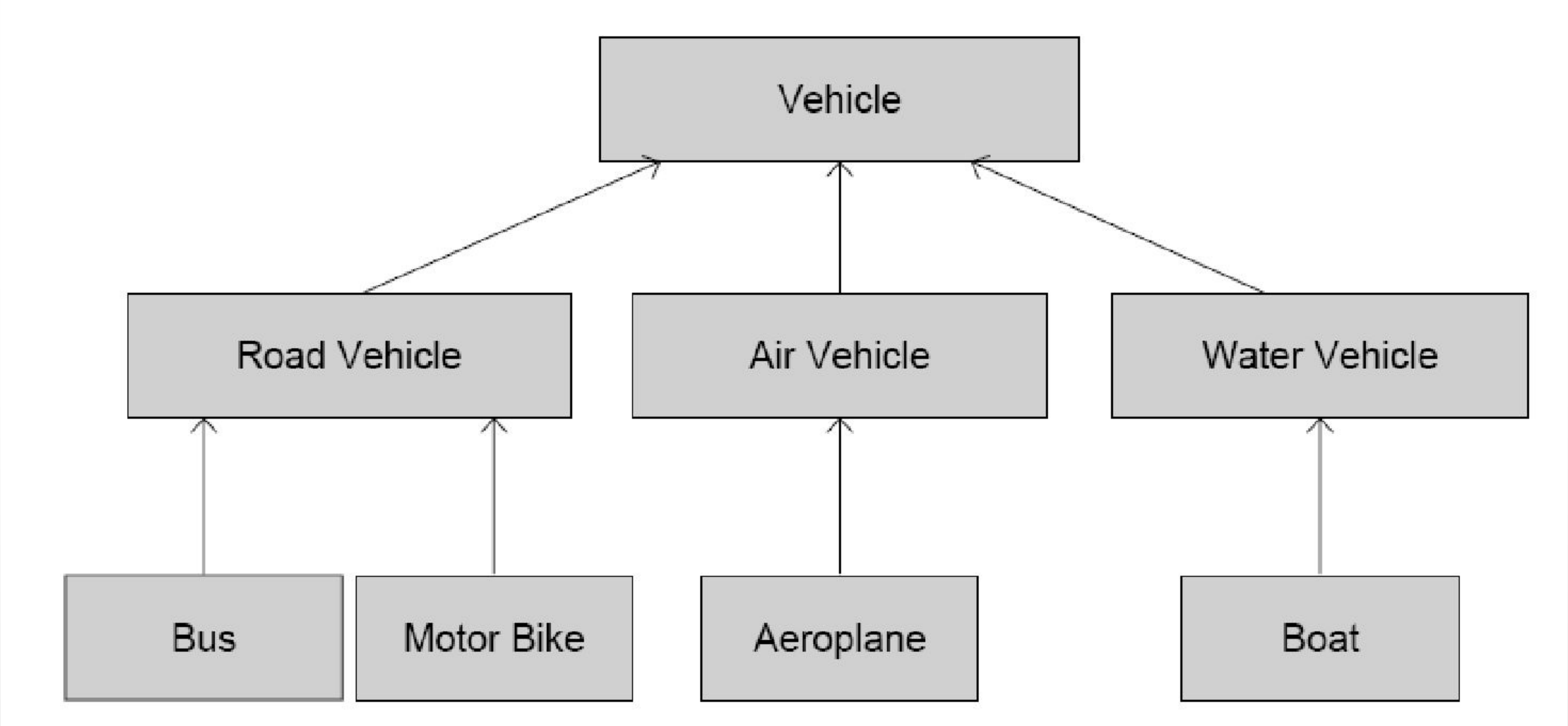
Abstraction

- In real life, humans manage complexity by abstracting details away.
- In programming, we manage complexity by concentrating only on the essential characteristics and suppressing implementation details.
- For example Car.

Inheritance

- Way to adopt characteristics of a class into another class.
- Have two types of classes: one is base class and other is subclass.
- A parent-child relationship among classes in inheritance.
- A subclass inherits all the properties of base class. In addition to this it can add its own features (properties and behavior).
- For example, we can categorize vehicle into car, bus, scooter, ships, planes etc.

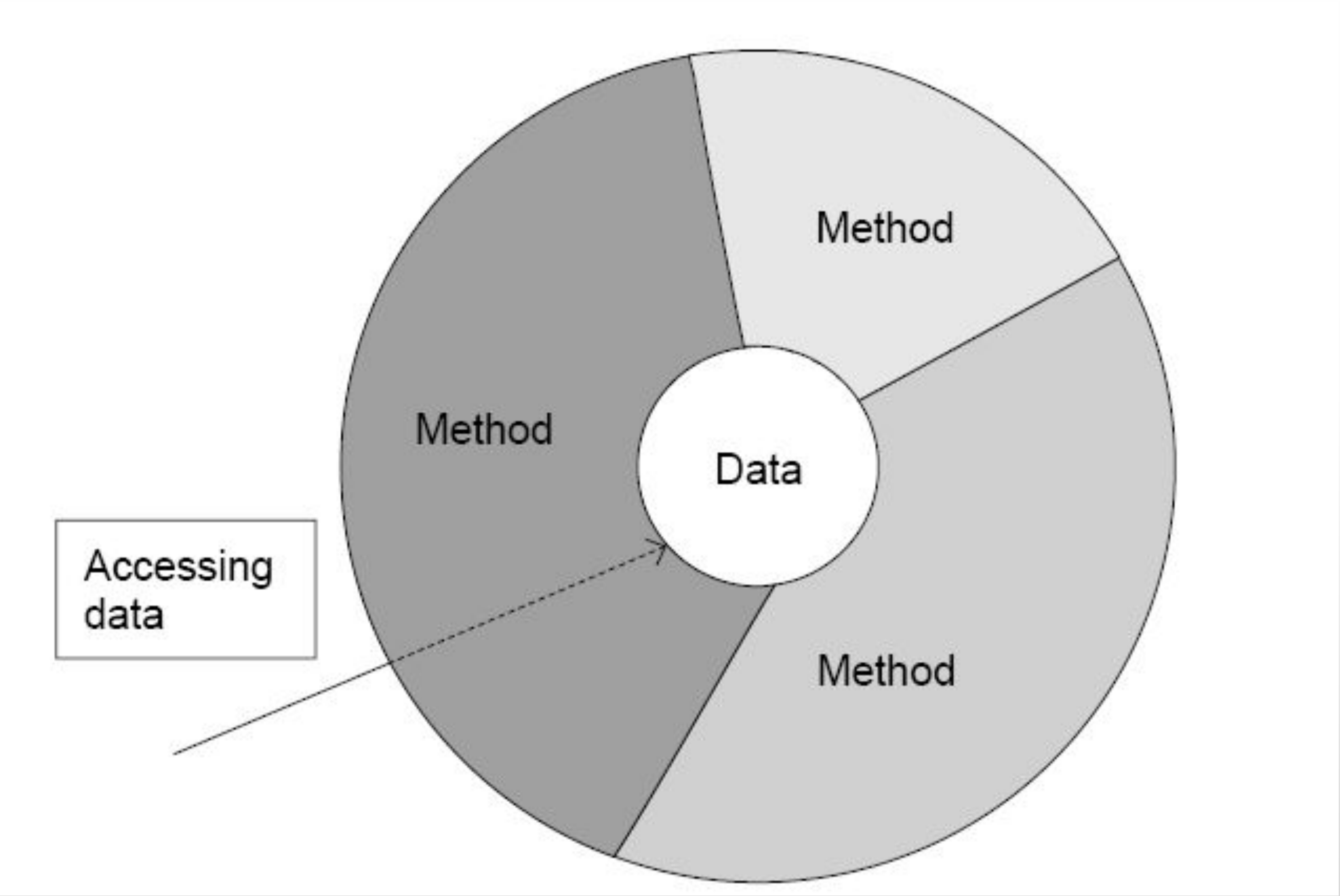
Inheritance



Encapsulation

- Binding of data and procedure.
- Restrict any one to directly alter our data.
- Also known as data hiding.

Encapsulation



Polymorphism

- Polymorphism means many forms.
- Same thing being used in different forms. For example certain bacteria that exhibit more than one morphological form.
- Operating overloading is not supported by java.
- Addition operator (**+**) is an exception can be used for addition of two integers as well as concatenation of two strings.
- Compile-time and run-time polymorphism.

Your Turn

- What is a class?
- What is an object?
- What are the various Object Oriented Principles?

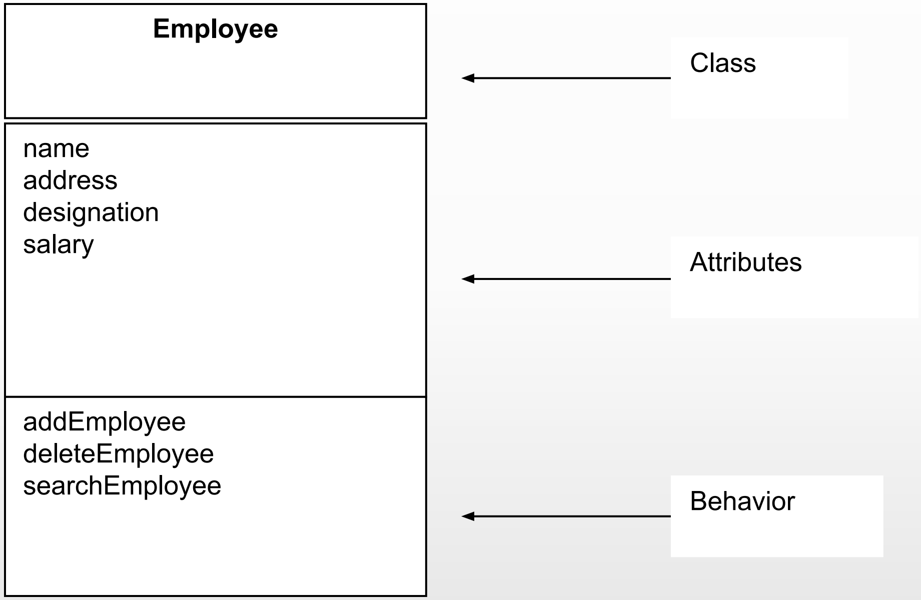
Comparison of OO and Procedural Languages

Procedural language	Object Oriented language
Separate data from function that operate on them	Encapsulate data and methods in a class
Not suitable for defining abstract types	Suitable for defining abstract types
Debugging is difficult	Debugging is easier
Difficult to implement change	Easier to manage and implement change
Not suitable for larger applications/programs	Suitable for larger programs and applications
Analysis and design not so easy	Analysis and Design Made Easier
Faster	Slower
Less flexible	Highly flexible
Data and procedure based	Object oriented
Less reusable	More reusable
Only data and procedures are there	Inheritance, encapsulation and polymorphism are key features
Use top down approach	Use bottom up approach
Only a function call to another	Object communication is there
C, Basic, FORTRAN	JAVA,C++, VB.NET, C#.NET

UML

- A standard language for OOAD.
- Graphical notation for all entities (class, object, etc.)
- Helps in visualizing the system.
- Reducing complexity and improving software quality.

UML Notation For class



Employee

name
address
designation
salary

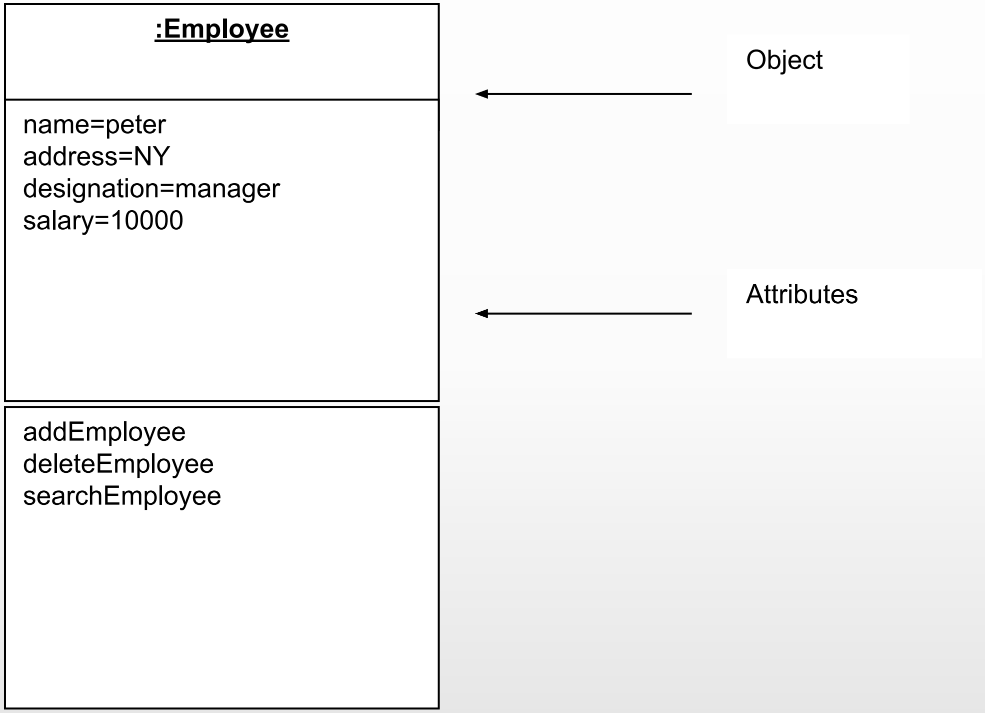
addEmployee
deleteEmployee
searchEmployee

Class

Attributes

Behavior

UML Notation for Object



:Employee

name=peter
address=NY
designation=manager
salary=10000

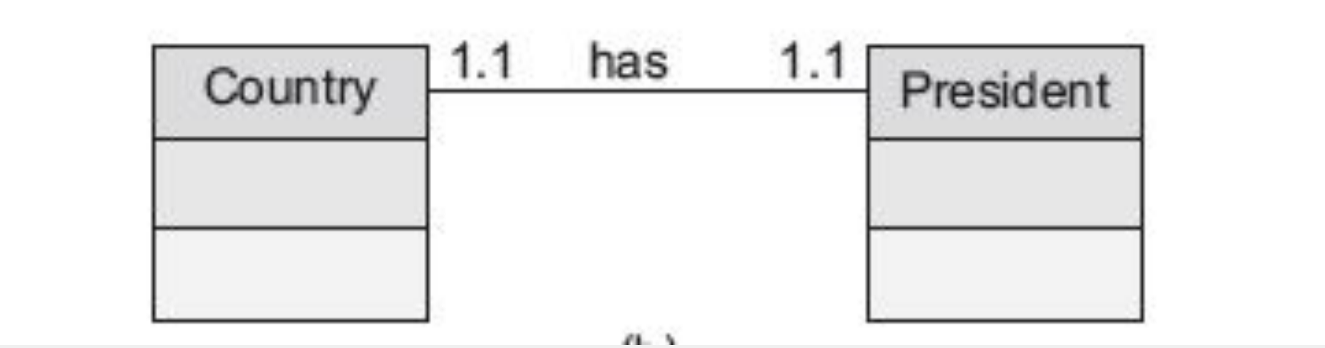
addEmployee
deleteEmployee
searchEmployee

Multiplicity

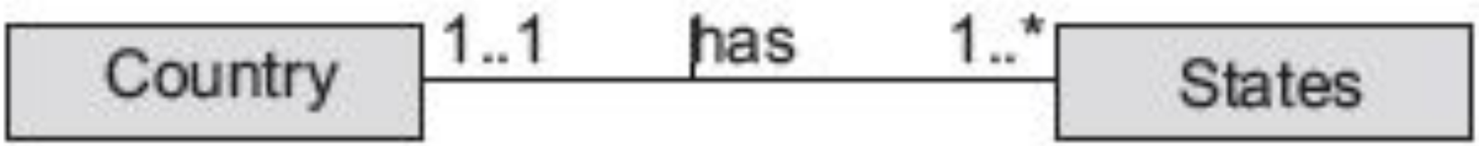
- An instance of a class can be related to any number of instances of other class known as multiplicity of the relation.

  - One-to-one
  - One-to-many
  - Many-to-many

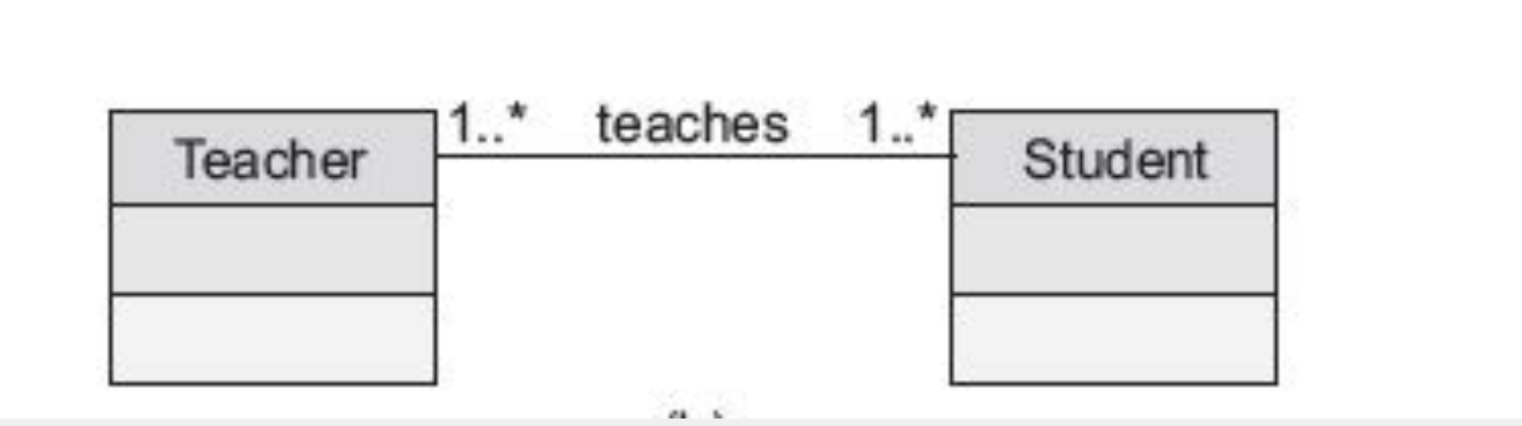
One-to-One relation



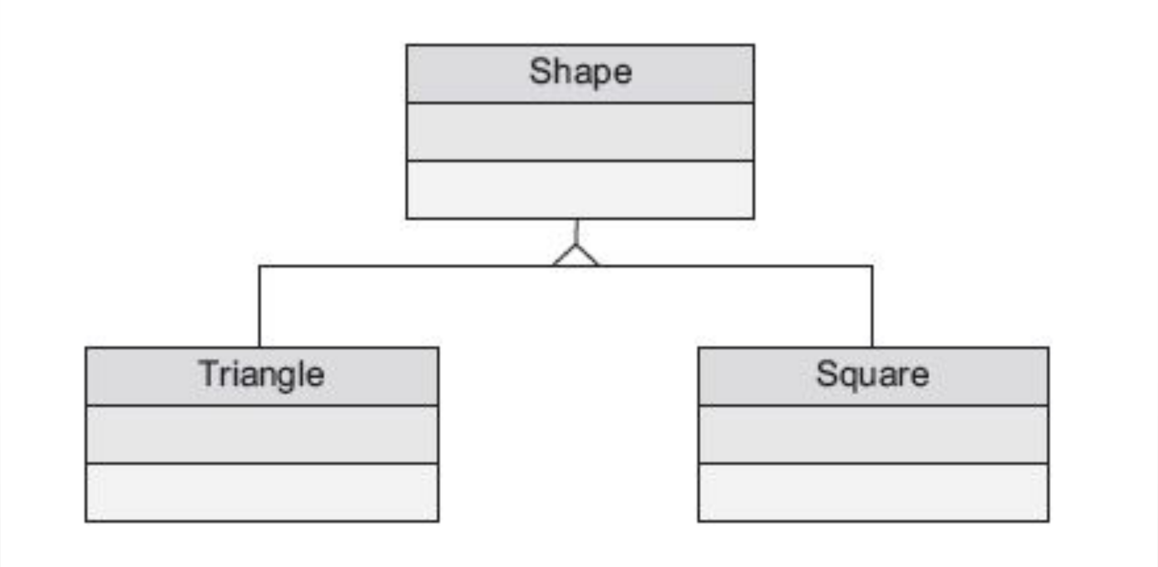
One to Many



Many to Many

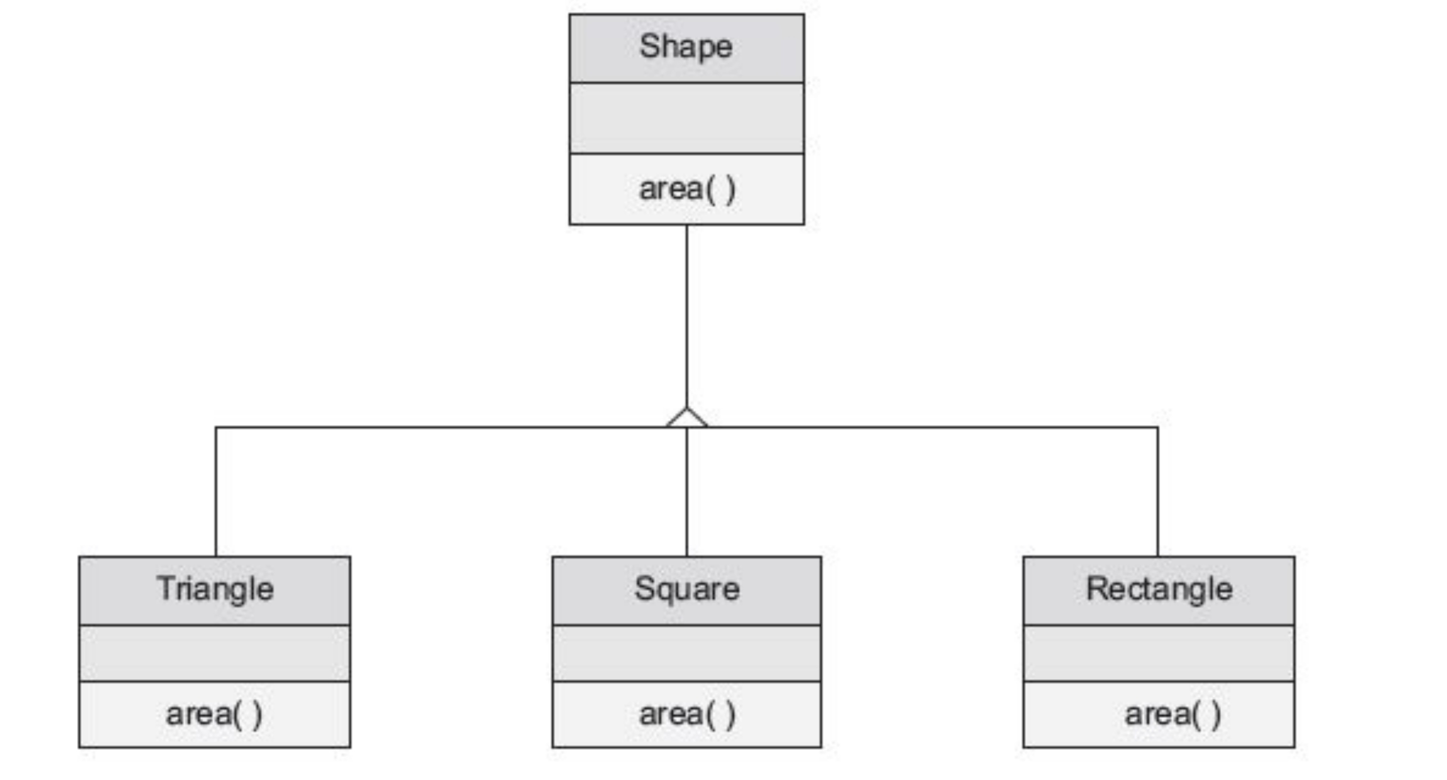


Notation for Inheritance

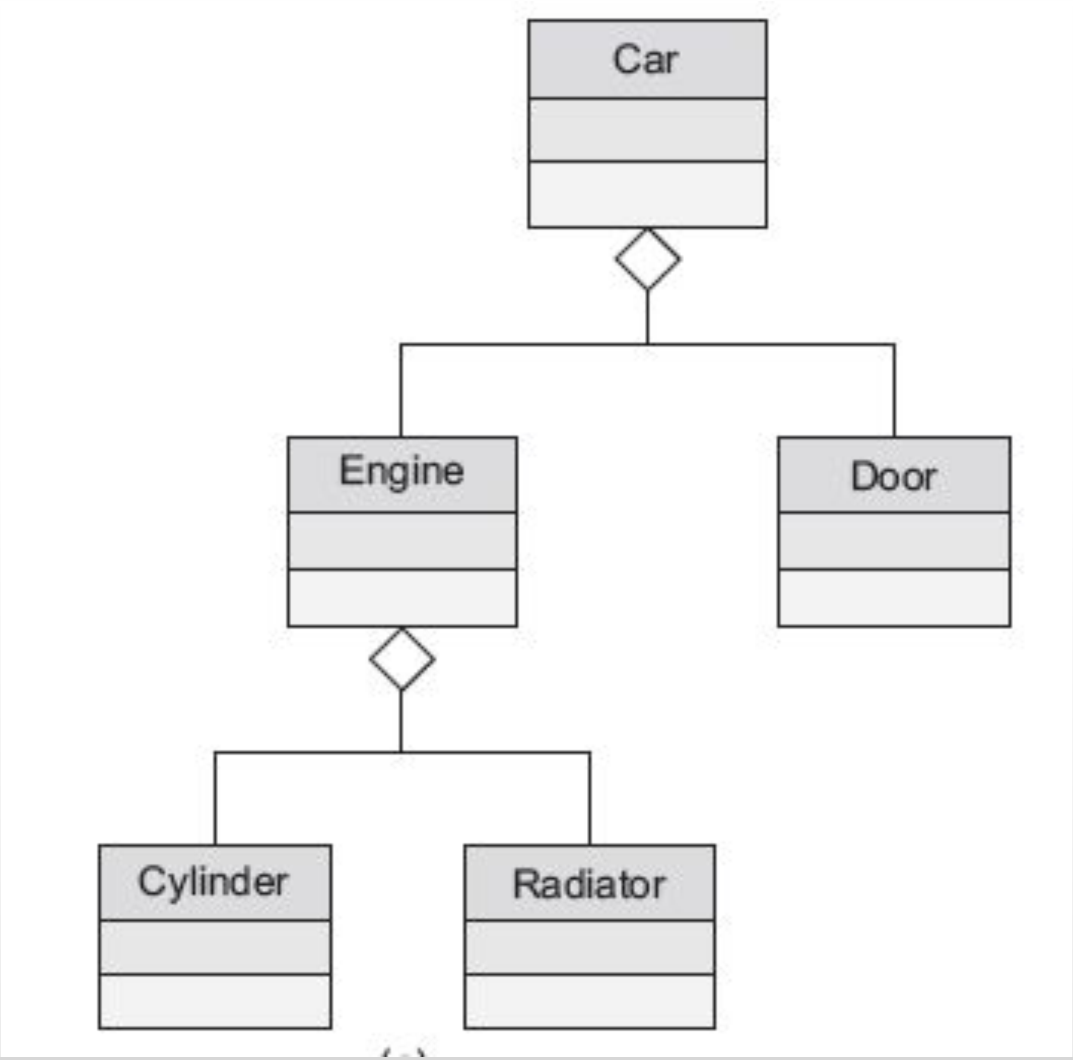


- Triangle and square are two type of shape.
- They inherit Shape class.

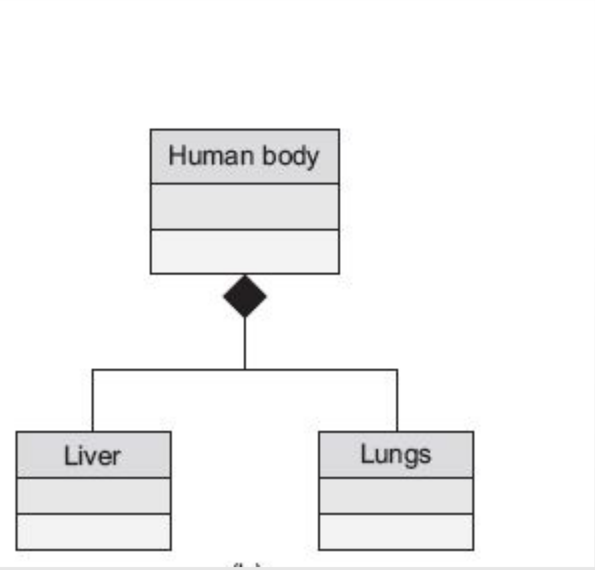
UML Diagram Depicting Polymorphism



Aggregation



Composition



Applications of OOP

- Real Time systems
- Artificial Intelligence
- Expert Systems
- Neural Networks
- Database Management
- Mobile Applications

Summary

- OOP revolves around objects and classes.
- A class is defined as a group of objects with similar attributes and behavior.
- OOP principles are Inheritance, Abstraction, Encapsulation and Polymorphism.
- For building large projects, a technique known as OOAD is used.
- OOA deals with what the system should do and OOD deals with how the system achieves what has been specified by OO Analysis.
- OOAD is realized with the help of a language known as UML.
- UML stands for Unified Modeling Language and is a standard language used for visualizing the software.