

MUST

Wisdom & Virtue

MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING

Object Oriented Programming

Lecture 8 : Inheritance in Object Oriented Model

Engr.Saman Fatima
Lecturer

- **Inheritance in OOP**
- **How to implement inheritance in C#**
- **Advantages of inheritance**

Last Lecture

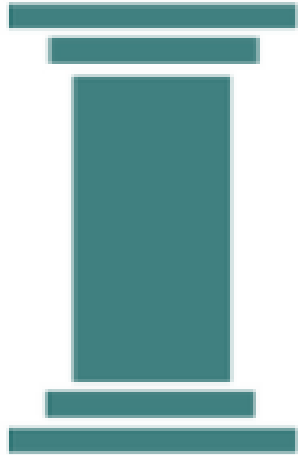
This Lecture

- **Why we use Inheritance**
- **Generalization**
- **Subtyping (extension)**
- **Specialization (restriction)**

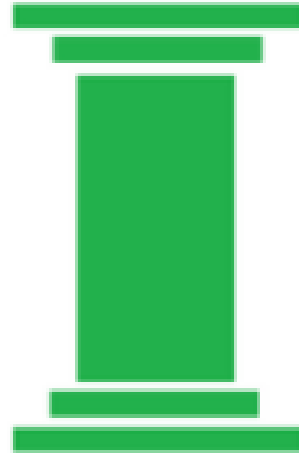


Four Pillars of OOP

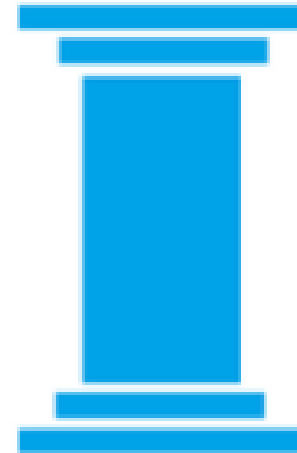
ABSTRACTION



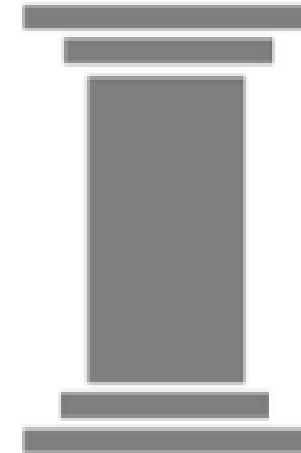
ENCAPSULATION




INHERITANCE



POLYMORPHISM





Why We use Inheritance (Purposes related to Inheritance)

Generalization

Subtyping (extension)

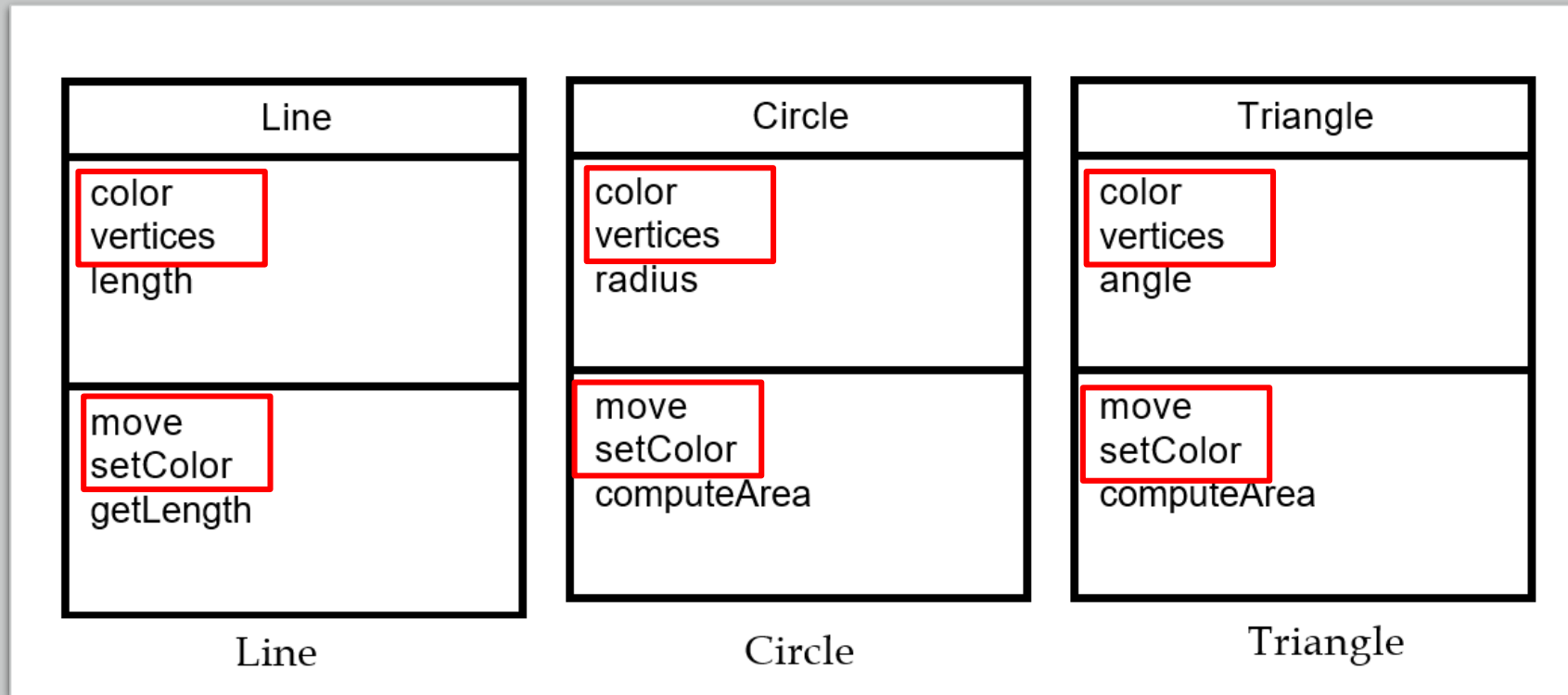
Specialization (restriction)

Generalization

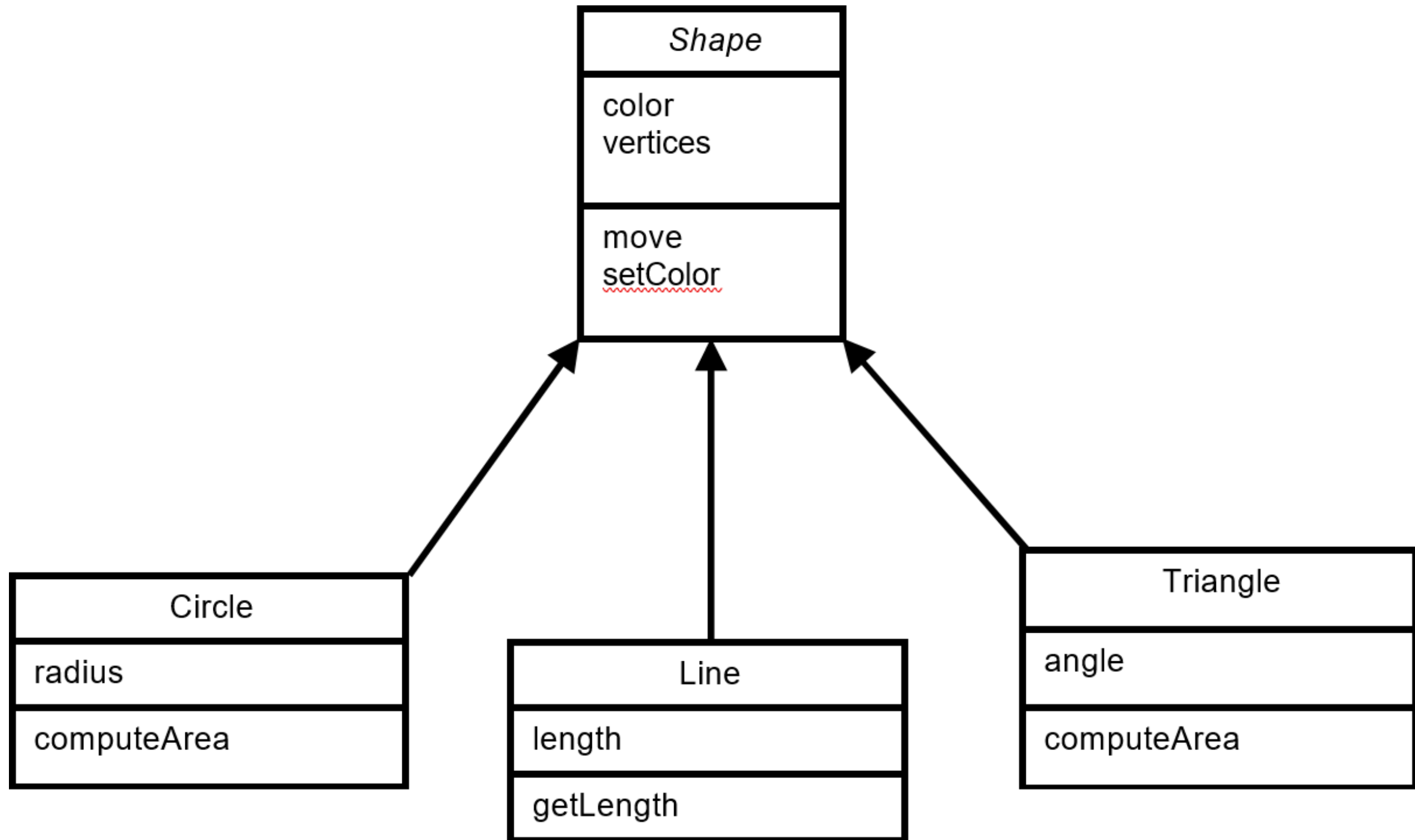
Generalization

- In OO models, some classes may have **common characteristics**.
- We extract these features into a new class and inherit original classes from this new class.
- There are many objects with common characteristics in object model.
- The common characteristics (attributes and behaviour) of all these objects are **combined in a single general class**.
- Base class encapsulates the idea of commonality of derived classes.
- Base class is general class representing common behaviour of all derived classes.
- This concept is known as **Generalization**.

Gernalization Focus on Common Attribute



Example: Line, Circle and Triangle

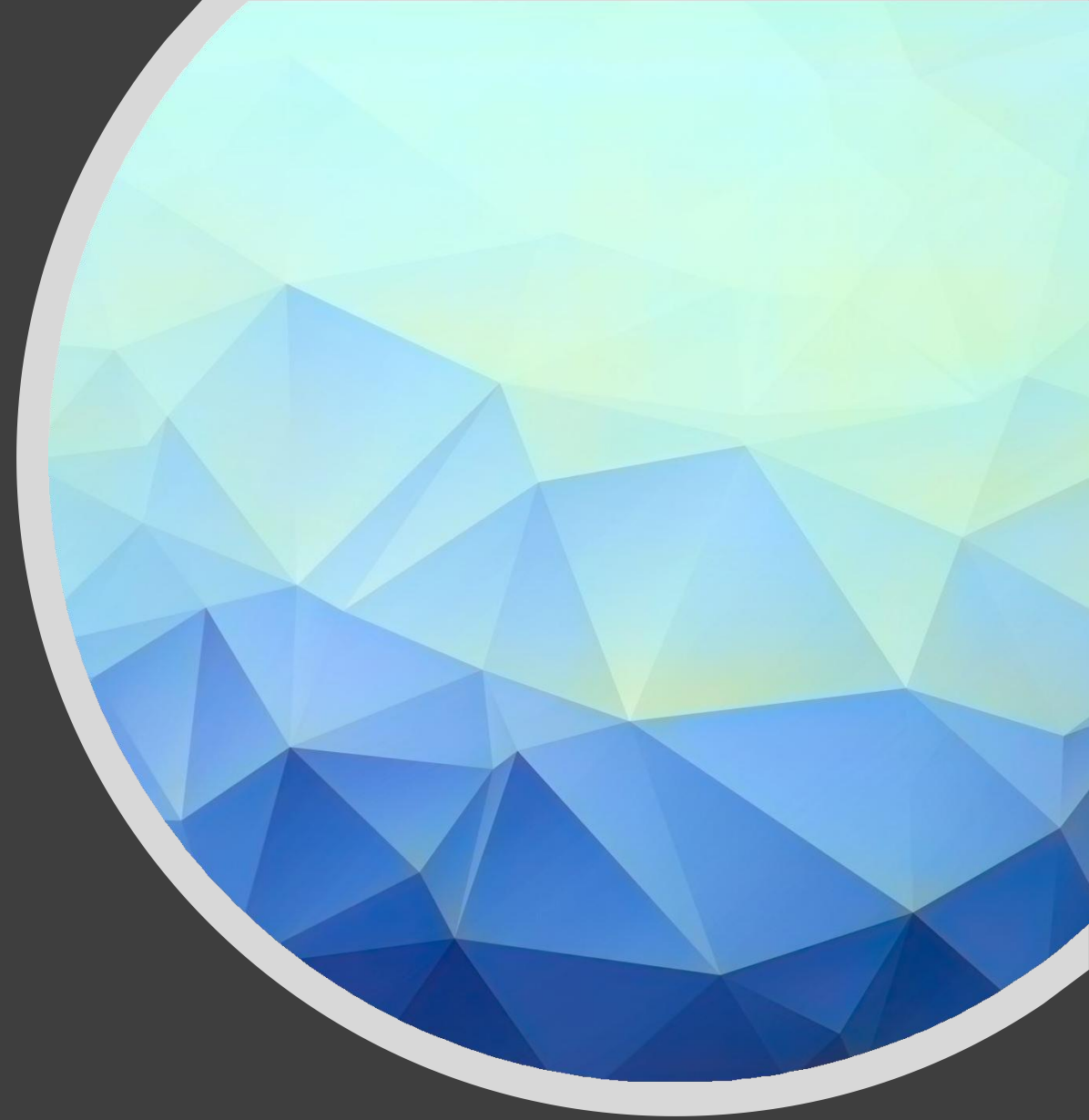


Shape will be the base class

Line is a
shape

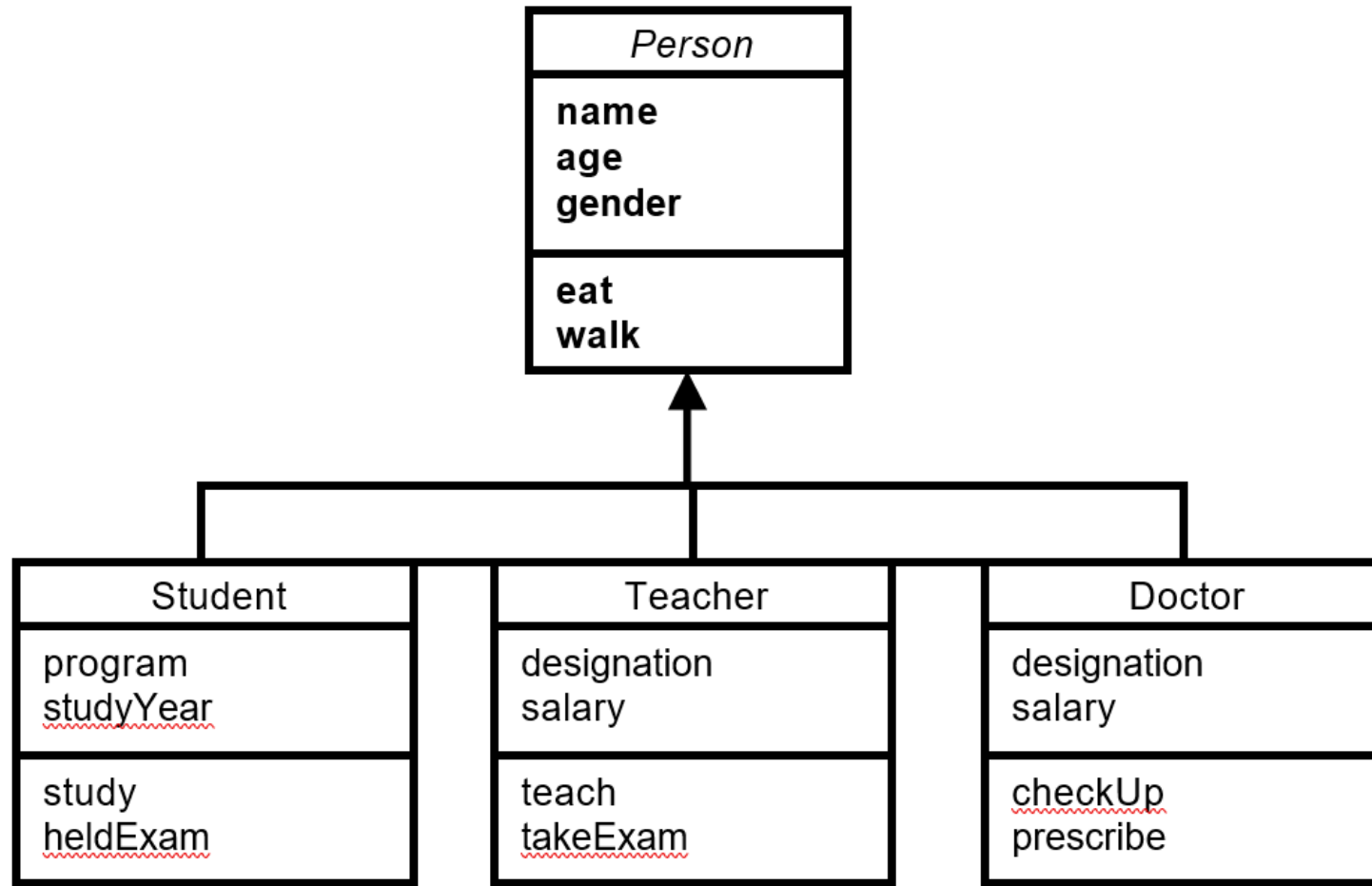
Circle is a
shape

Triangle is
a shape



Student	Teacher	Doctor
<div>name age gender</div> program studyYear	<div>name age gender</div> designation salary	<div>name age gender</div> designation salary
study heldExam <div>eat walk</div>	teach takeExam <div>eat walk</div>	checkUp prescribe <div>eat walk</div>

Example: Student Doctor and Teacher



Common attributes,
Name, age, gender

Common behaviour
Eat, Walk

Advantage of Generalization

It reduces the **redundancy** and gives us **reusability**, using generalization our solution becomes **less complex**.

In generalization there should be “**Is a** Kind of Relationship” (also called “Is A relationship”) between base and child classes.

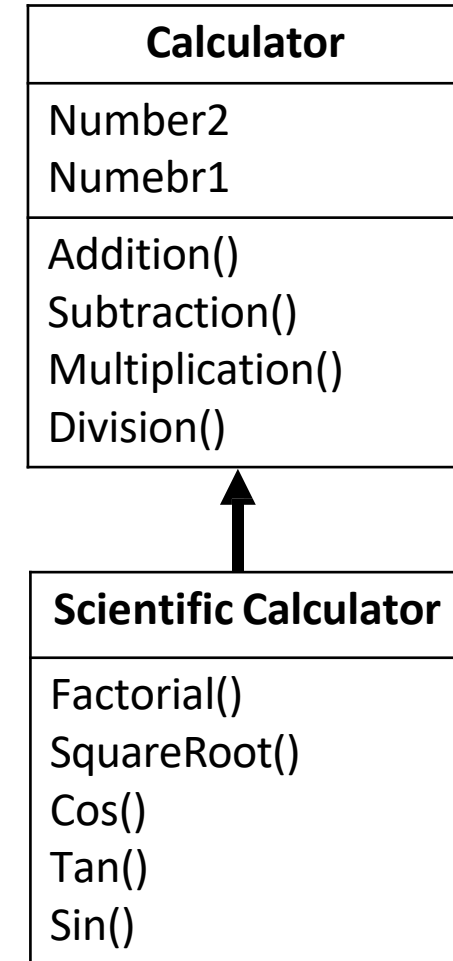
Sub-typing

1. Sub-typing (Extension)

- Sub-typing means that **derived class** is behaviorally compatible with the **base class**
- Derived class has all the characteristics of base class plus some **extra characteristics**

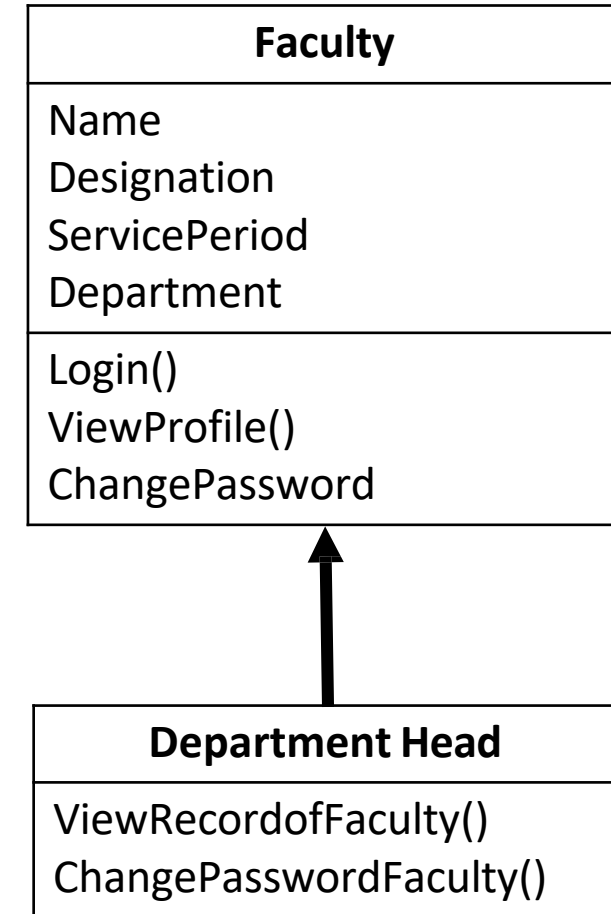
Sub-typing (Extension) – Example-3

- Scientific Calculator has **extended** behavior of Calculator by adding **Factorial, SquareRoot, Cos, Tan** and **Sin** functionality.



Sub-typing (Extension) – Example-3

- Department Head has **extended** behavior of Faculty by adding ViewRecordofFaculty and ChangePasswordFaculty functionality.

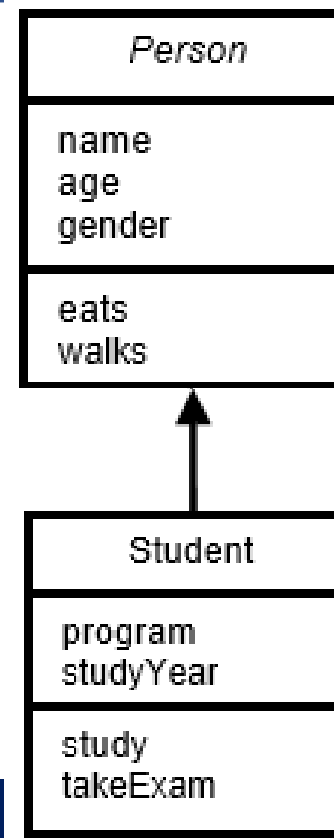


Generalization VS Sub-typing

Subtyping and Generalization are Related Concepts

- Subtyping (extension) and generalization is a way to look same thing in two ways.

Sub typing is looking at things from **Top to bottom**



Subtyping and Generalization are Related Concepts

- Subtyping (extension) and generalization is a way to look same thing in two ways.

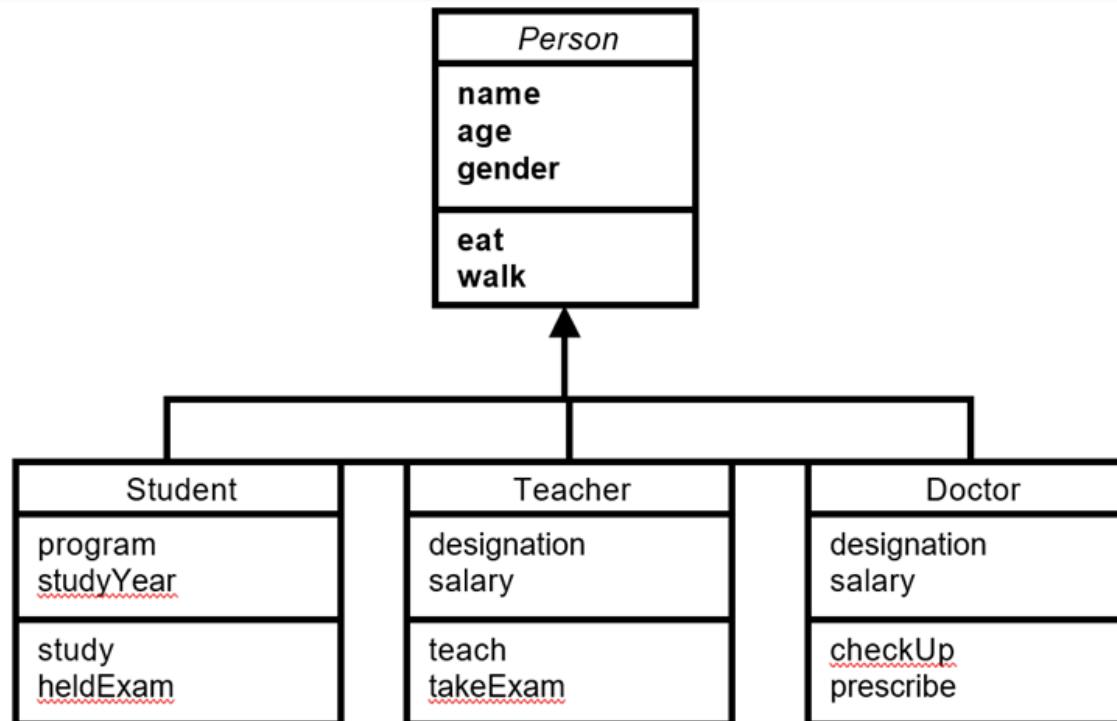
Generalization we look at things from **bottom to top**

Student	Teacher	Doctor
<div>name age gender</div> <div>program studyYear</div>	<div>name age gender</div> <div>designation salary</div>	<div>name age gender</div> <div>designation salary</div>
<div>study heldExam</div> <div>eat walk</div>	<div>teach takeExam</div> <div>eat walk</div>	<div>checkUp prescribe</div> <div>eat walk</div>

Subtyping and Generalization are Related Concepts

- Subtyping (extension) and generalization is a way to look same thing in two ways.

Generalization we look at things from **bottom to top**



Specialization

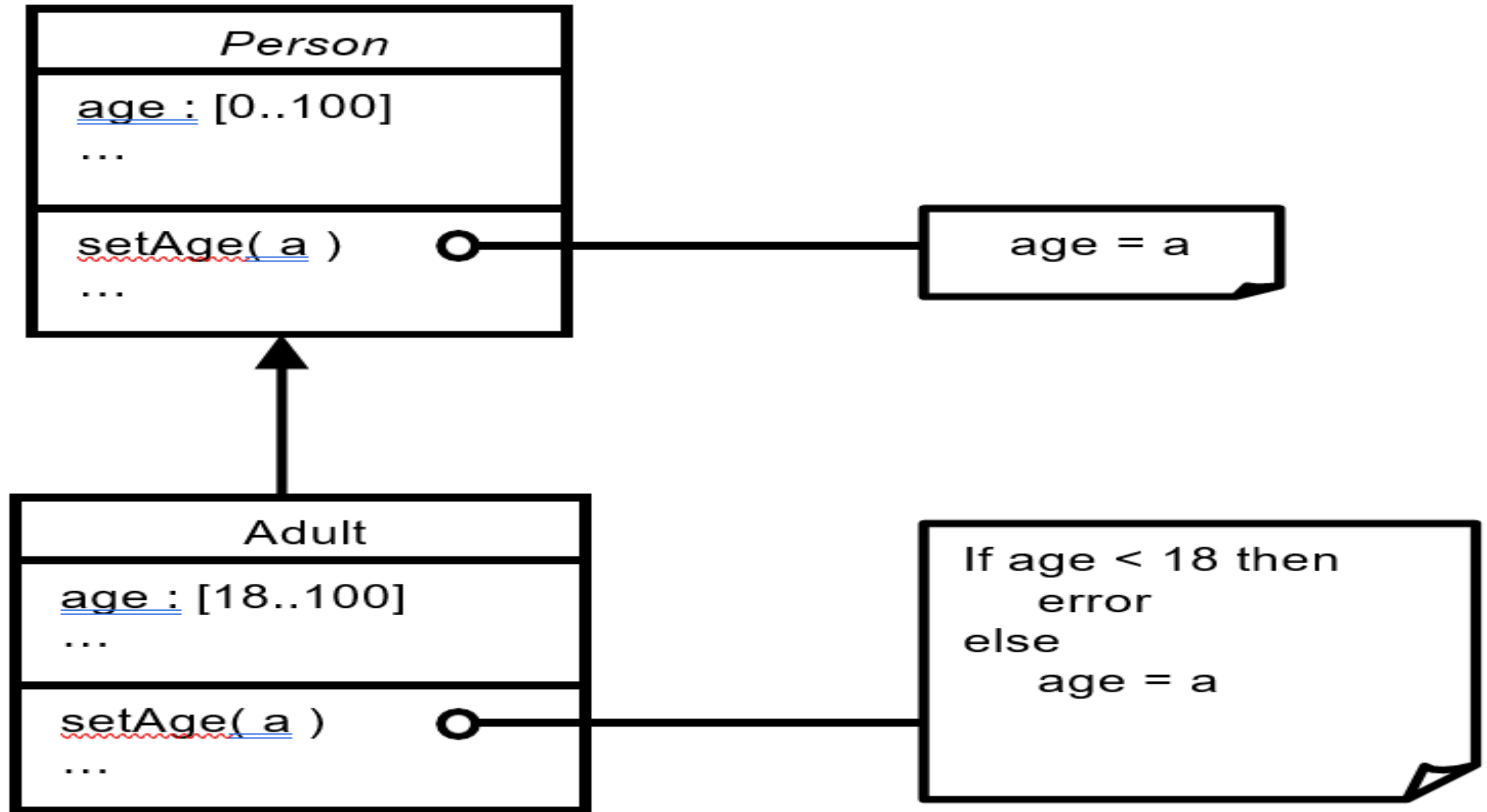
Specialization (Restriction)

- We want to add a class to **existing hierarchy** of classes having many **similarities** to already existing classes but some part of its behaviour is different or **restricted**.
- In that case we will use the concept of specialization.
- Specialization means that derived class is **behaviourally incompatible** with the base class
- Behaviourally incompatibility means that base class can't always be replaced by the derived class
- Derived class has some different of restricted characteristics than of base class.

Example – Specialization (Restriction)

Suppose we want to add one more class of **Adult** for some special requirement like for ID card generation such that it is a person but its age is greater than 18 and having all other behaviour of that of person class.

One solution is that we write another class from beginning and write all code of person again in it with age limit, but **better solution** is that we derive adult class from person class and restrict age in that class as shown in diagram on next slide

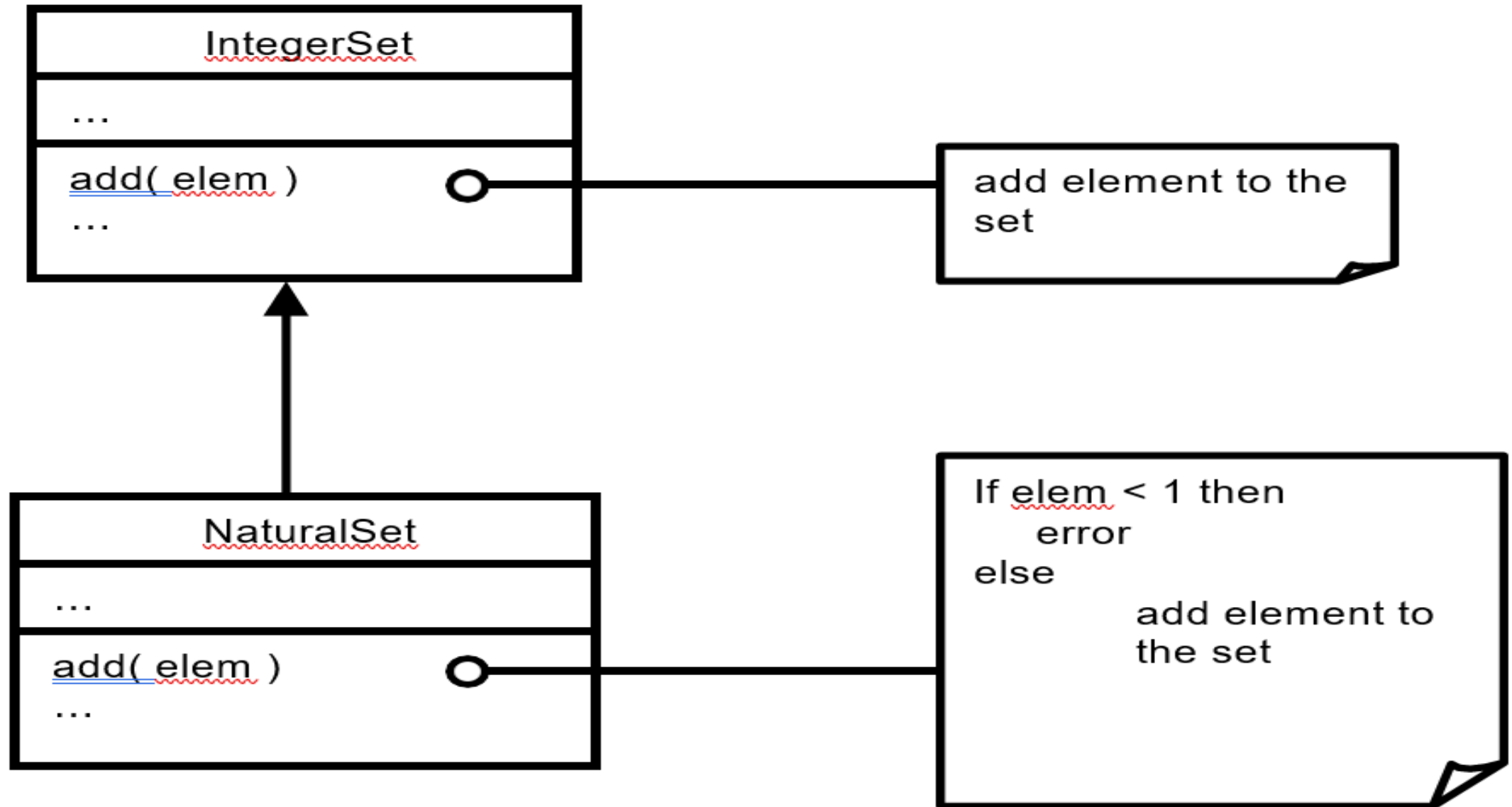


Example – Specialization (Restriction)

Similarly, Natural Numbers are also Integers with the restriction that natural numbers set can **NOT contain zero or negative integers** it consists of only positive integers so we can implement this relationship also as specialization

Natural numbers: positive integers only (numbers from 1 toonwards)

Integers: all positive and negative numbers (.....-3 , -2 , -1 , 0 , 1 , 2 , 3.....)



References

- Object Oriented Programing , Virtual University , Lecture 4, Online
Available at:
<https://ocw.vu.edu.pk/CourseDetails.aspx?cat=Computer+Science%2FInformation+Technology+&course=CS304>

THANKS