Monsoon 2019

Association, Aggregation, and Composition

Object Oriented Programming

by

Dr. Rajendra Prasath

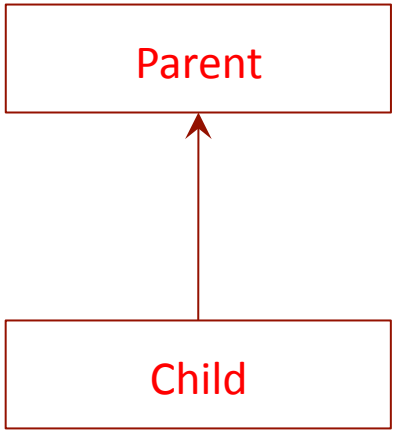
Indian Institute of Information Technology

Sri City – 517 646, Andhra Pradesh, India

Recap: Objects in JAVA ?

- An entity that has **state** and **behaviour** is known as an object
  - **Examples:** Chair, bike, marker, pen, table, car etc
  - It can be physical or logical
- An object has three characteristics:
  - **State:** represents data (value) of an object
  - **Behaviour:** represents the behaviour (functionality) of an object such as deposit, withdraw and so on
  - **Identity (Internally used):**
    - Signature (unique) of the object
    - Object identity is typically implemented via a unique ID
    - The value of the ID is not visible to the external user
    - But, Internally by JVM to identify each object uniquely

Recap: Class Casting



✧ Let us look at the following:

Create an Object:

Parent p = new Parent();

Upcasting:

Parent p = new Child();

Downcasting:

Child c = new Parent() ➔ Compile time error

Child c = (Child) new Parent();

➔ This throws Class CastException at run time.

➔ How to solve this issue?

Polymorphism

- Polymorphism in java is a concept by which we can perform a single action by different ways
- Polymorphism is derived from: **poly** and **morphs**
- The word "**poly**" means many and "**morphs**" means forms → So polymorphism means many forms

- **Example:**
  - Person in a class room – Faculty / Student
  - Person in market – Customer
  - Person at home – Son or Daughter
  - Person at Work – Founder / Employee

    → one person present in different behaviors

Association

What is an Association?

- Objects interact with each other
- Usually an object provides services to other objects
- An object keeps associations with other objects to delegate tasks

Class Association

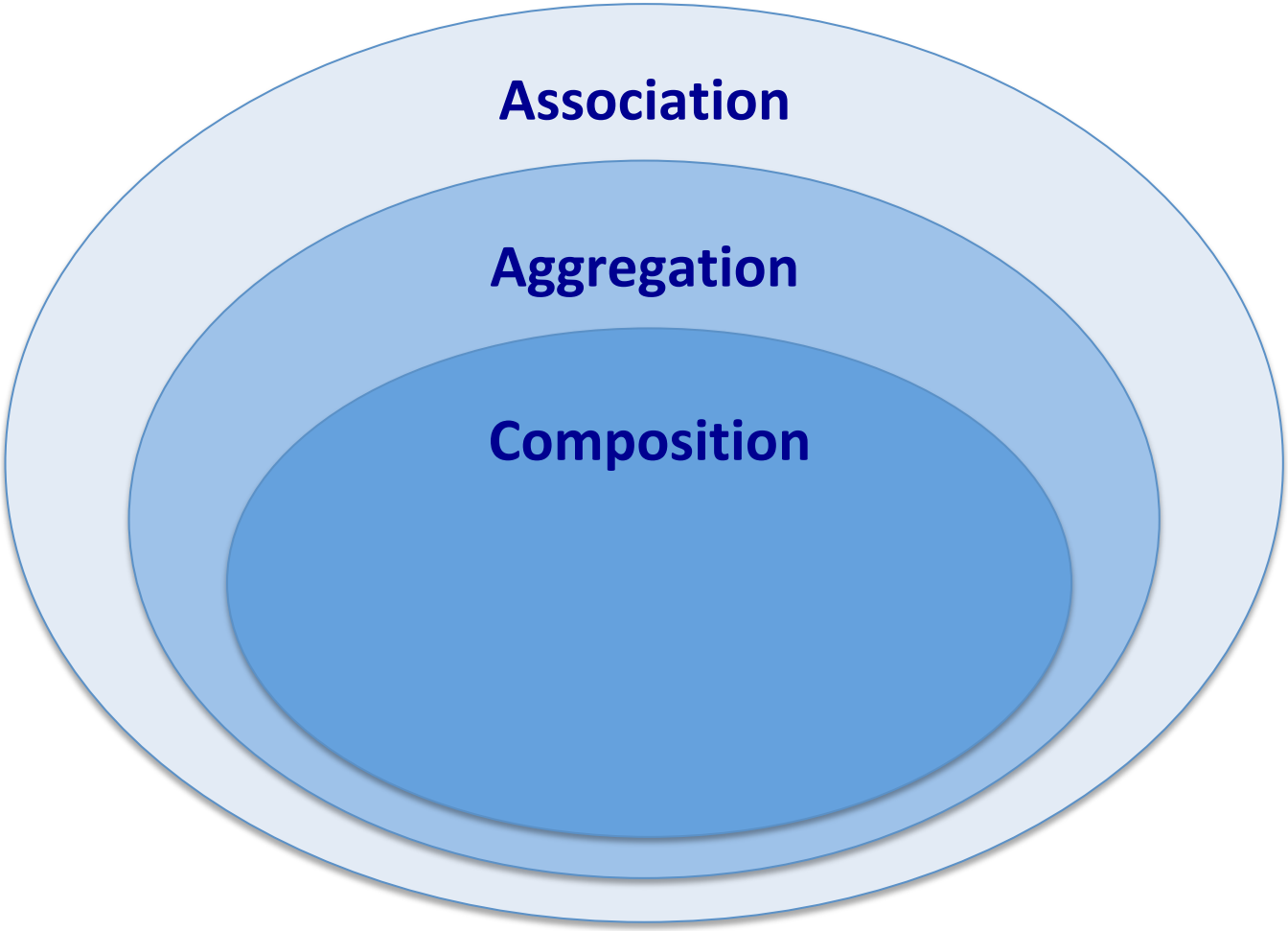
- Inheritance

Object Association

- Simple Association
- Composition
- Aggregation

Simple Association

Let us look at the following:



Simple Association

- Is the weakest link between objects
- Is a reference by which one object can interact with some other object
- Is simply called as "Association"

Ways:

- One way association
- Two way association

Number of Objects

- Binary Association
- Ternary Association
- N-ary Association

One-Way Association

Navigate along a single direction only

✧ Denoted by an arrow towards the server object

Example:

✧ Ram lives in a house

✧ Ram drives his car

Two-way Association

- Navigates in both directions
- Denoted by a line between the associated objects

Example:

- Ram works for company
- Company employs employees

- Ram is a friend of Rahim
- Rahim is a friend of Ram

Binary Association

- Associate objects of exactly two classes
- Denoted by a line or an arrow between the associated objects

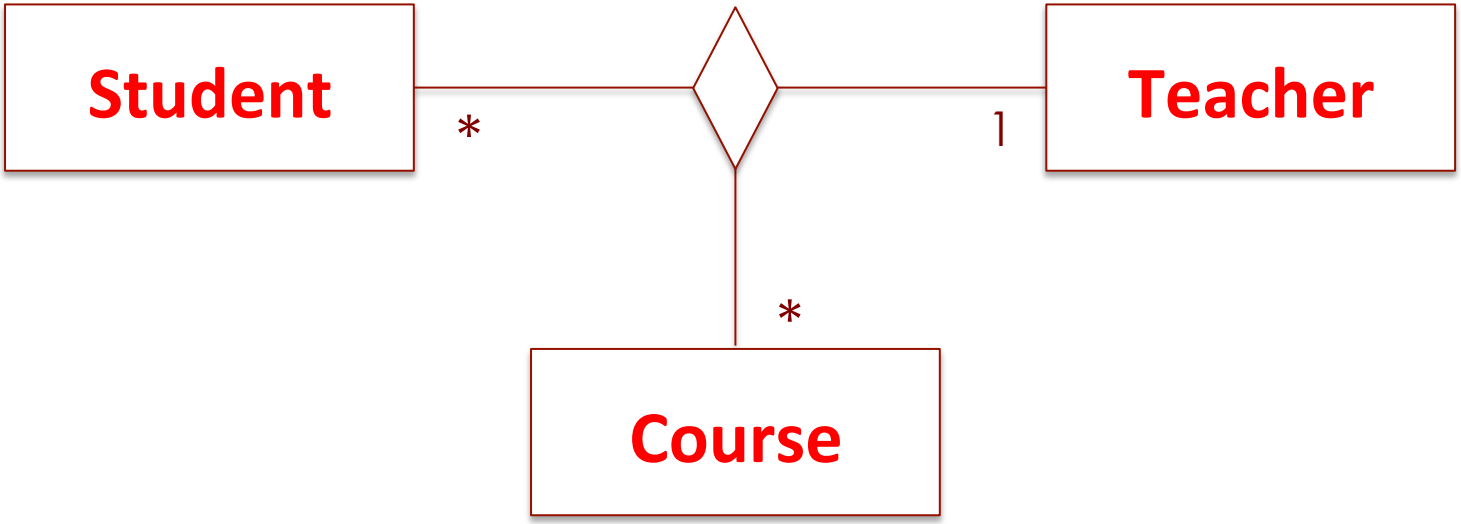
Example:

- "Works for" associates objects of exactly two classes

Ternary Association

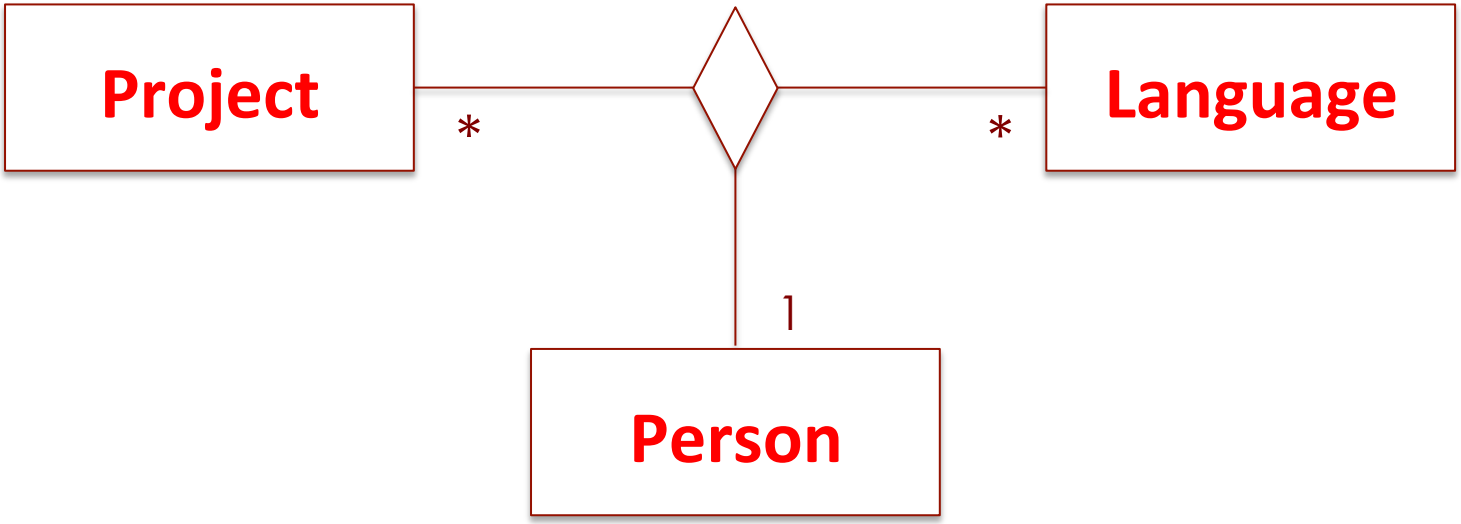
- Associates objects of exactly three classes
- Denoted by a diamond with lines connected to Associated objects

Example:



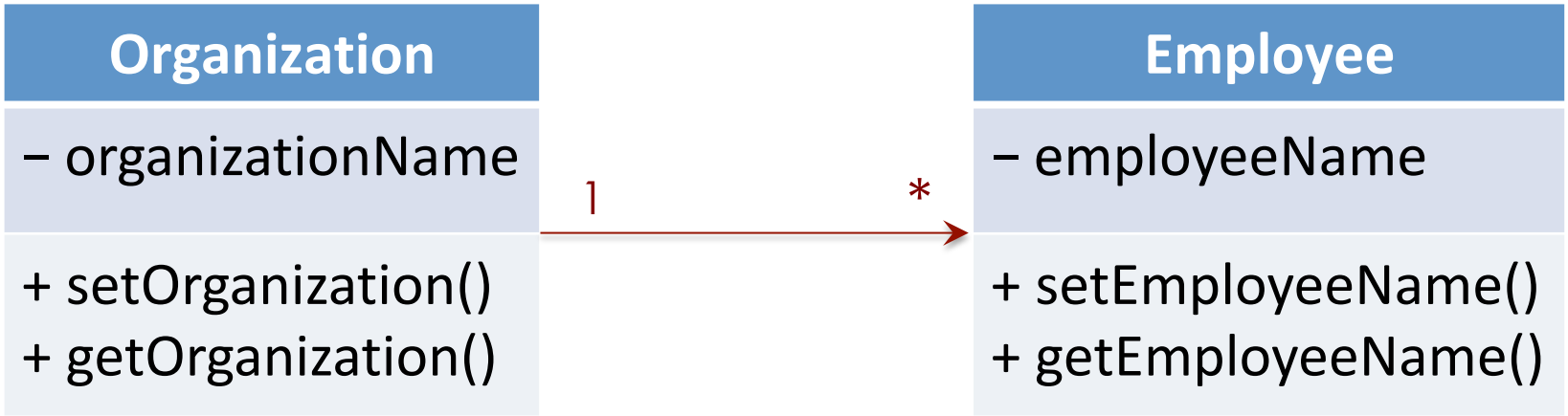
Example - Ternary Association

- Objects of exactly three classes are associated



Class Diagram - Association

- Example: Organization and Employee have an association
- Let us look into the following:
- The minus sign – shows that the field is **private**
- The plus sign – shows that the field is **public**



Organization – Employee

✧ Look at the following code:

```
class Organization {
    private String orgName;
    public String getOrgName() { return orgName;}
    public void setOrgName(String orgName) { this.orgName = orgName; }
}

class Employee {
    private String empName;
    public String getEmpName() { return empName; }
    public void setEmpName(String empName) { this.empName = empName; }
}

public class CodeChecker {
    public static void main(String args[]) {
        Organization org = new Organization(); org.setOrgName("2power3.com");
        Employee emp = new Employee();
        emp.setEmpName("Dr. Kathik");
        System.out.println(emp.getEmpName() + " is a staff of " + org.getOrgName());
    }
}
```

Car - Driver

```
class Car {
    public String name;
    public double speed;
    int ID;
    Car(String name, double speed, int id) {
        this.name = name; this.speed = speed; this.ID = id;
    }
}

class Driver {
    public String driverName;
    public int driverAge;
    Driver(String name, int age) { this.driverName = name; this.driverAge = age;}
}

class CodeChecker {
    public static void main(String[] args) {
        Car car = new Car("BMW", 100.0, 3323); Driver driver = new Driver("Karthik", 45);
        System.out.println(driver.driverName + " is a driver of car ID: " + car.ID);
        System.out.println("The speed of " + car.name + " car is " + car.speed + " per hour");
    }
}
```

Aggregation

What is an Aggregation?

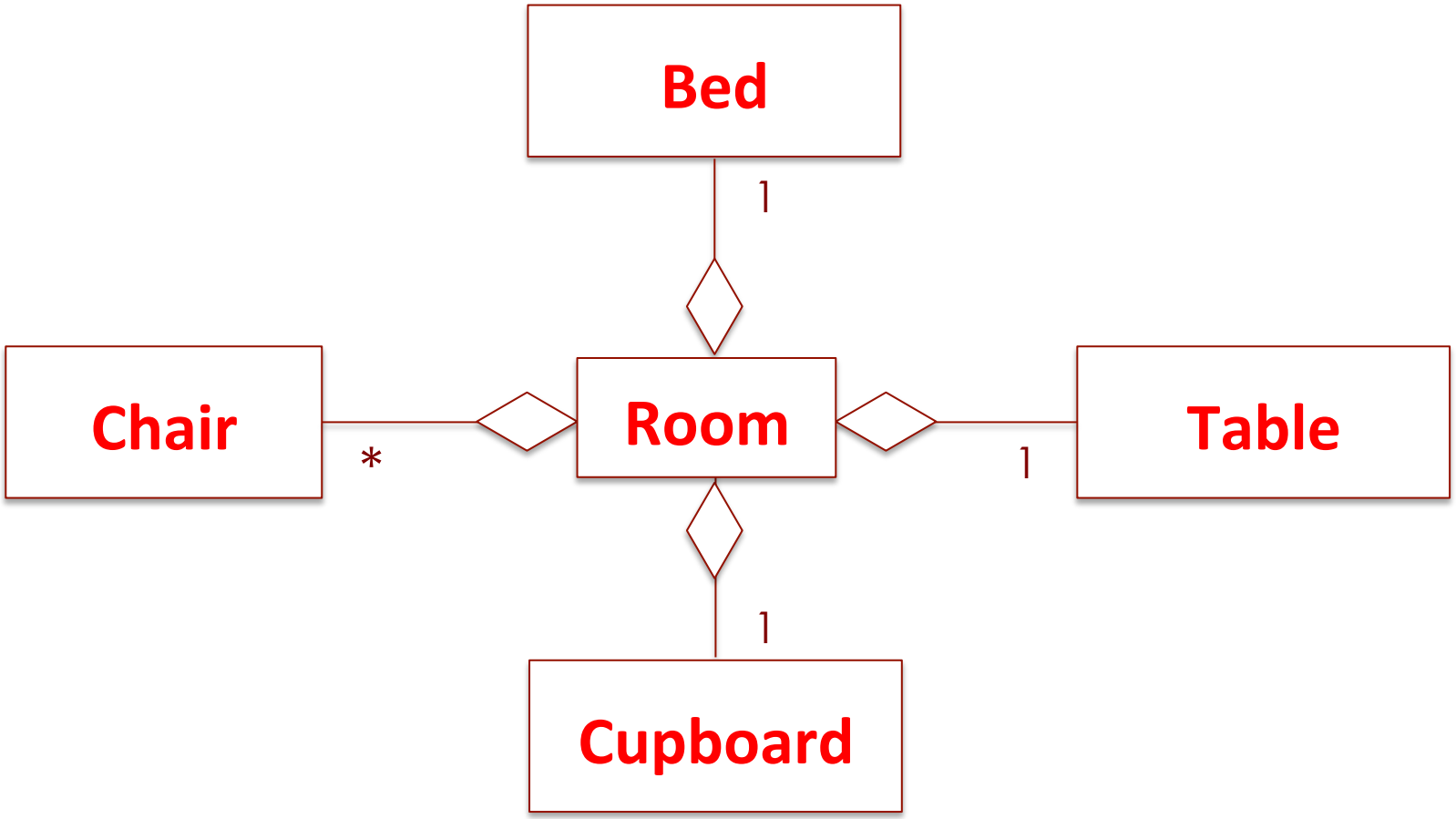
- An Object may contain a collection of other objects
- The relationship between the container and the contained object is called an Aggregation
- Aggregation is represented by a line with unfilled-diamond head towards the container
- Aggregation is a specialized form of Association

Example:

- Department and Teacher
- "has-a" relationship

Example - Aggregation

- Let us look at an example



Aggregation - Properties

- Aggregation is a weaker relationship
  - Aggregate object is not a part of the container
  - Aggregate object can exist independently

Example:

- Furniture can be shifted to another room and so it can exist independent of a particular room

Aggregation in JAVA

✧ Let us look at an example

```
class Chair {
        public void show() {
                System.out.println("Show Chair Details");
        }
}

public class Room {
        public void static main(String args[]) {
                Chair ch = new Chair();
                ch.show();
        }
}
```

Composition

What is an Composition?

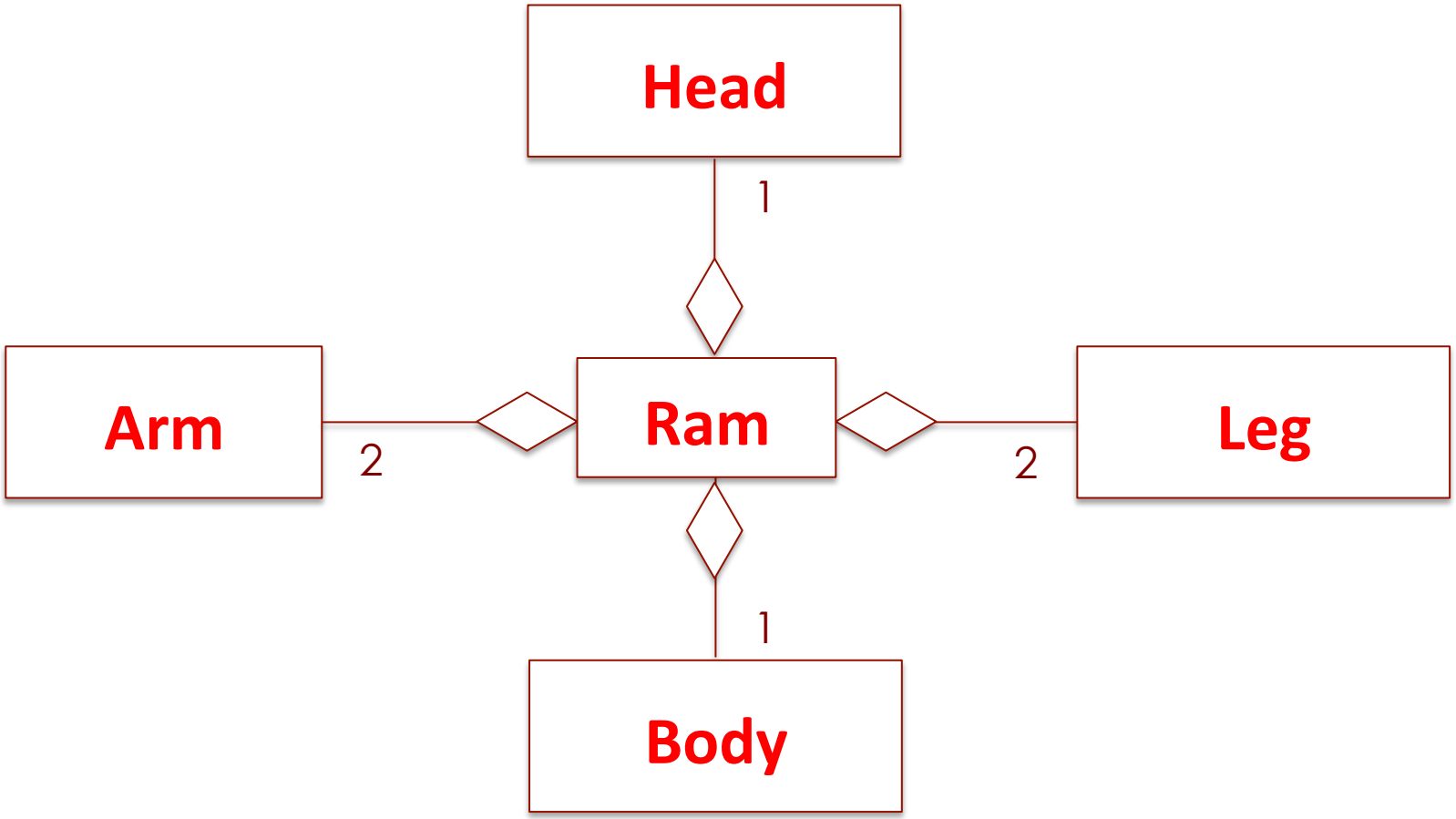
- An object may be composed of other smaller objects
- The relationship between the "part" objects and the "whole" object is known as Composition
- Composition is represented by a line with a filled-diamond head towards the composer object
- It is also called "has-a" relationship
- We can reuse the functionality of another class

Example:

- **House** can contain multiple **Rooms**
- **Room** is not independent without **House**

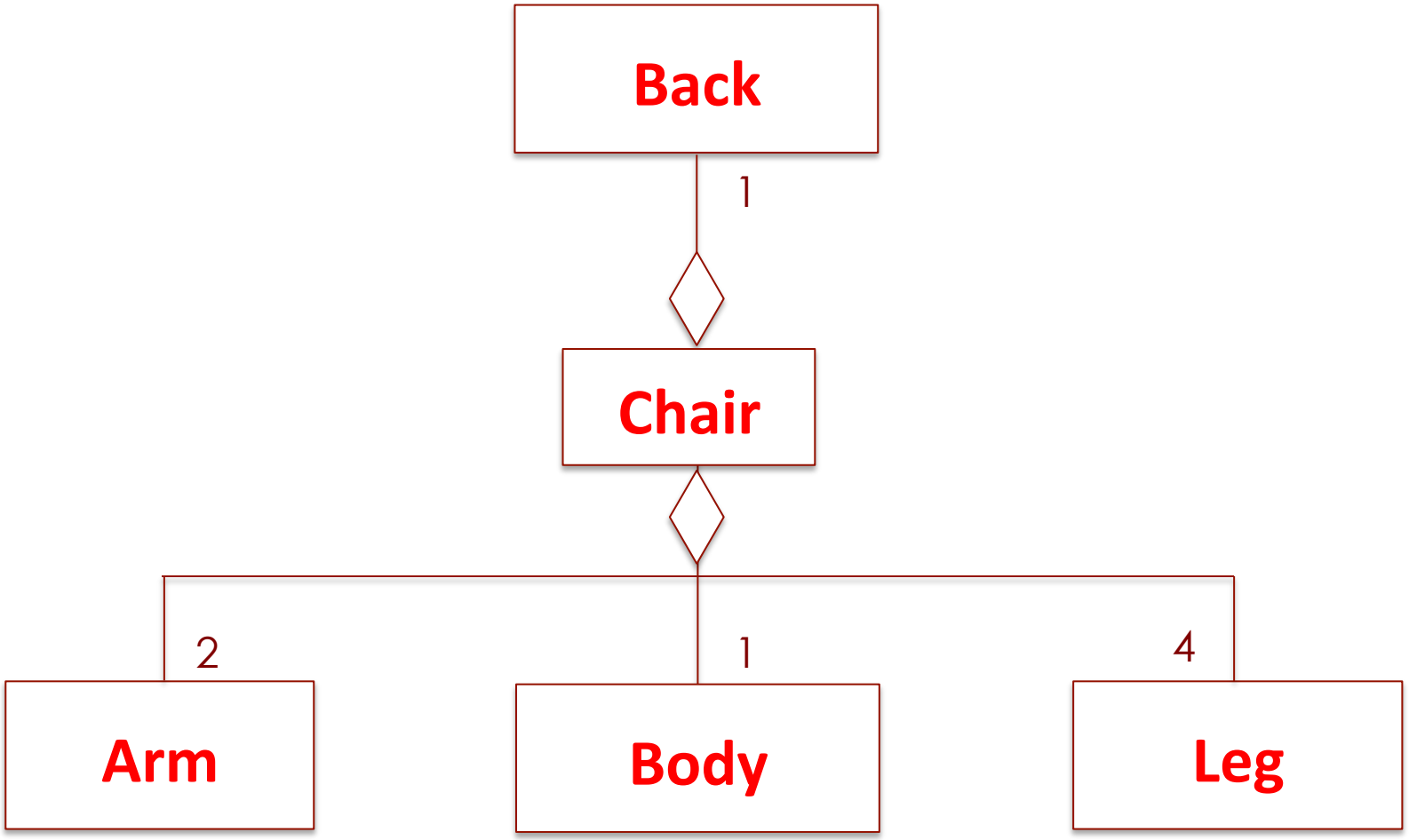
Example -Composition

✧ Let us look at an example



Example -Composition

- Let us look at an example



Syntax - Composition

- When a class having attribute of another class then it is called composition

- Let us look at the syntax

```
class A {
        // code
}

class B {
        private A a;          // Composition
}
```

Composition between classes

- Example:
  - House – Room: "has – a" relationship
  - A house is composed of one or more rooms.
  - If house object is destroyed, all rooms are destroyed

```
public class House {
        private Room room;

        public house () {
                room = new Room();
        }
}
```

Staff – Hiring and Date

```
class Date {
        private int day, month, year;
        Date(int dd, int mm,
                this.day = dd; th
        }
        public String toString(
                return (day + "/
        }
}
class Staff {
        private int id; private
        private Date hireDat
        Staff(int num, String na, Date hire) {
                this.id = num; this.name = na; hireDate = hire;
        }
        public void display() {
                System.out.println("ID: " + id + "\nName: " + name
                        + "\nHiringDate: "+hireDate);
        }
}
```

```
public class CodeTester {
    public static void main(String[] args) {
            Date date = new Date(10, 17, 2000);
            Staff staff = new Staff(11, "John", date);
            staff.display();
    }
}
```

Exercise - 9

- **Perform Various Garbage Collection Algorithms**
  - Mark and Sweep
  - Collector Based GC
  - Age Based GC

Assignments / Penalties

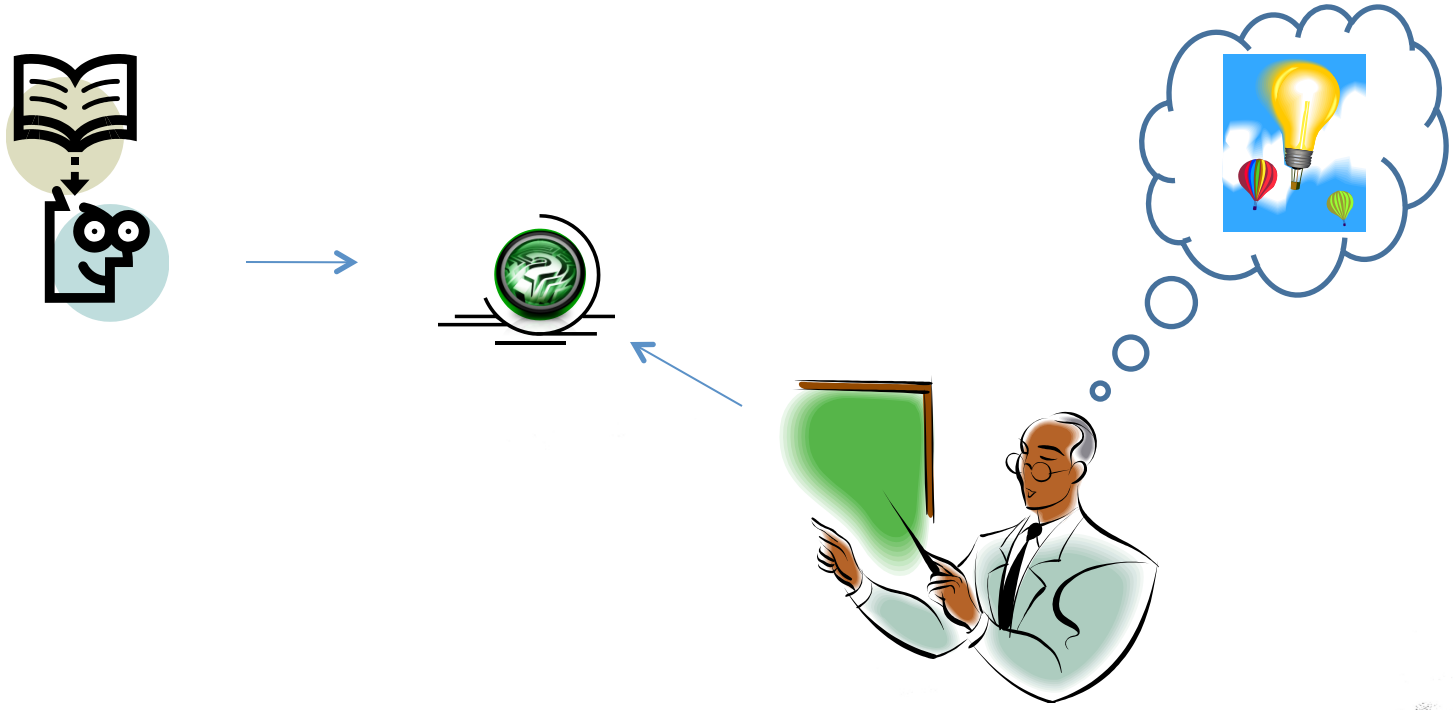
- Every Student is expected to complete the assignments and strictly follow a fair Academic Code of Conduct to avoid severe penalties

- Penalties would be heavy for those who involve in:
  - **Copy and Pasting** the code
  - **Plagiarism** (copied from your neighbor or friend – in this case, both will get "0" marks for that specific take home assignments)
  - If the candidate is **unable to explain his own solution**, it would be considered as a "copied case" !!
  - **Any other unfair means** of completing the assignments

Assistance

- You may post your questions to me at any time
- You may meet me in person on available time or with an appointment
- You may leave me an email any time
  (email is the best way to reach me faster)

Thanks …



… Questions ???