



Object-Oriented Programming (CS F213)

Module I: Object-Oriented and Java Basics

CS F213 RL 6.2: Aggregation and Composition

BITS Pilani

Dr. Pankaj Vyas

Department of Computer Science, BITS-Pilani, Pilani Campus

CS F213 RL 6.2 : Topics

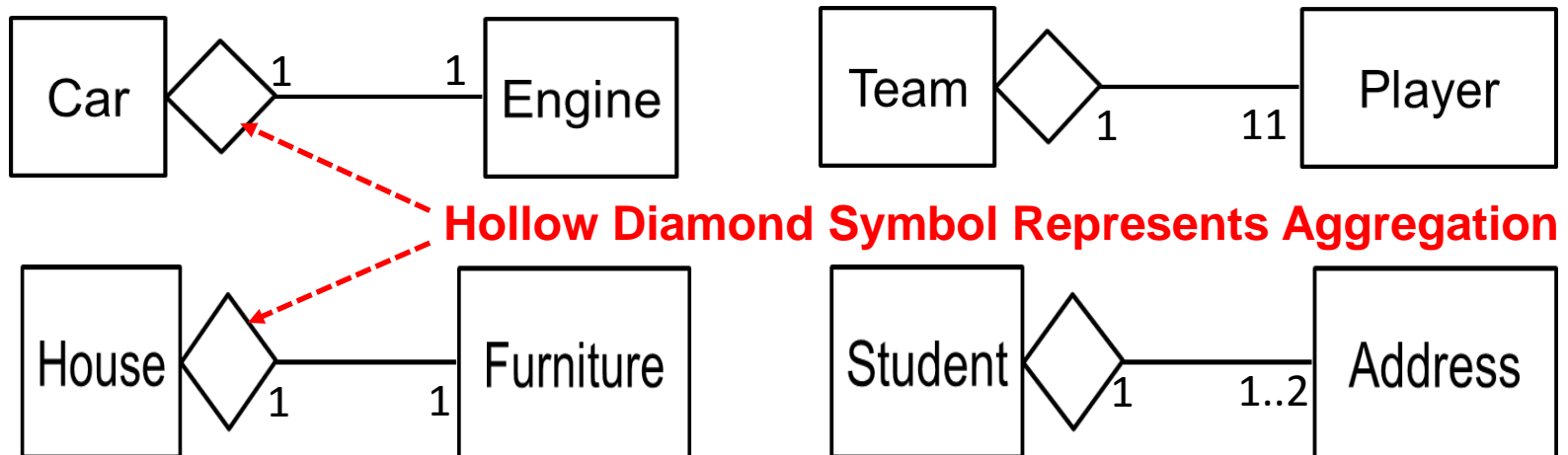


- Aggregation
- Composition

Aggregation



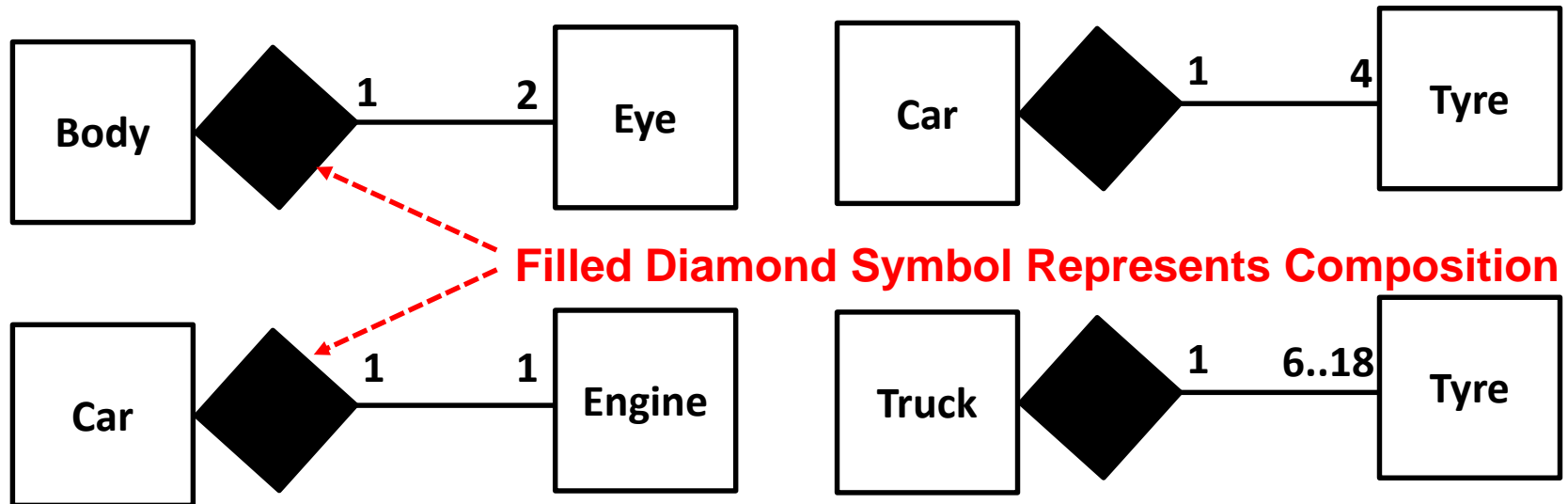
- Represents 'hasA' or 'a-part-of' relationship among classes and objects
- 'Car' has one 'Engine' (Alternatively 'Engine' is a-part-of 'Car')
- 'House' has 'Furniture' (Alternatively 'Furniture' is a-part-of 'House')
- 'Team' has 11 'Player' (Alternatively 'Player' is a-part-of 'Team')
- 'Student' has 'Address' (Alternatively 'Address' is a-part-of 'Student')



Composition



- Strong form of aggregation or ownership
- Whole object (Also known as Composite) can not exists without its participant objects
- 'Body' has two 'Eye'
- 'Car' has Four 'Tyre'



Aggregation vs Composition

Example 1



```

class Engine { } // End of class Engine
class Tyre   { } // End of class Tyre
class Car
{
    private Engine    engine;
    private Tyre      tyre-1;
    private Tyre      tyre-2;
    private Tyre      tyre-3;
    private Tyre      tyre-4;

    // Mutator Method for engine
    public void        setEngine(Engine eng)
    {
        this.engine = eng;
    } // End of Method

    // Mutator Method for tyre-1
    public void        setTyre1(Tyre t)
    {
        this.tyre-1 = t;
    } // End of Method

```

```

// Mutator Method for tyre-2
public void          setTyre2(Tyre t)
{
    this.tyre-2 = t;
} // End of Method

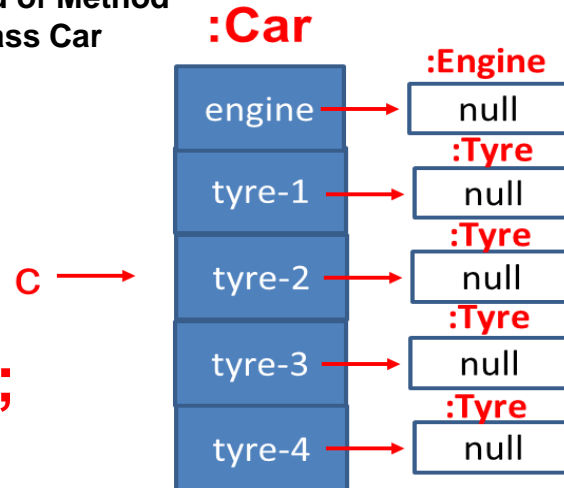
// Mutator Method for tyre-3
public void          setTyre3(Tyre t)
{
    this.tyre-3 = t;
} // End of Method

// Mutator Method for tyre-4
public void          setTyre4(Tyre t)
{
    this.tyre-4 = t;
} // End of Method

} // End of class Car

```

Car c = new Car();

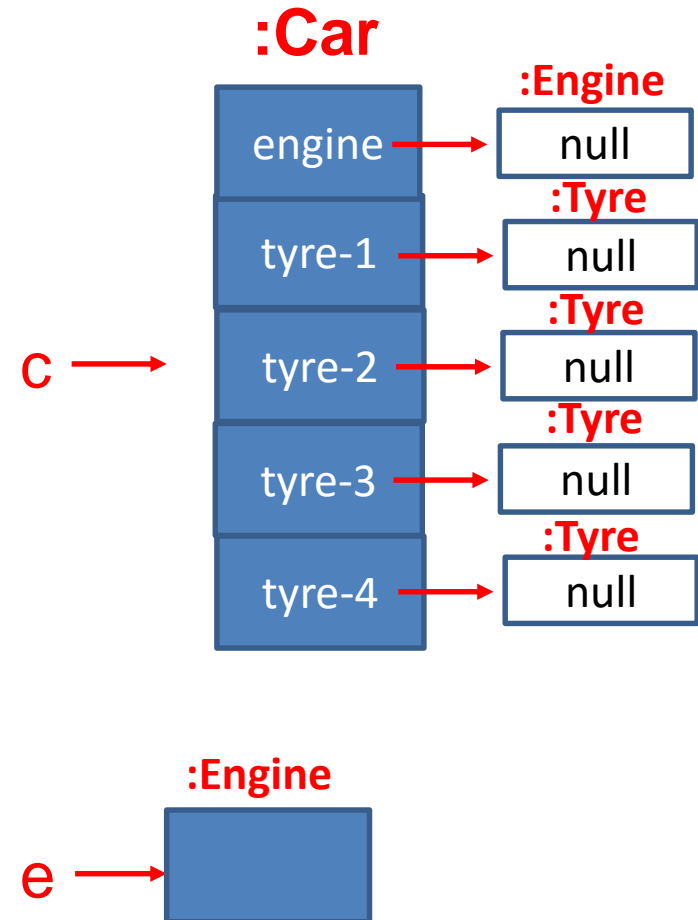


Aggregation vs Composition

Example 1



```
Car    c    =    new    Car();  
Engine e    =    new    Engine();  
c.setEngine(e);
```



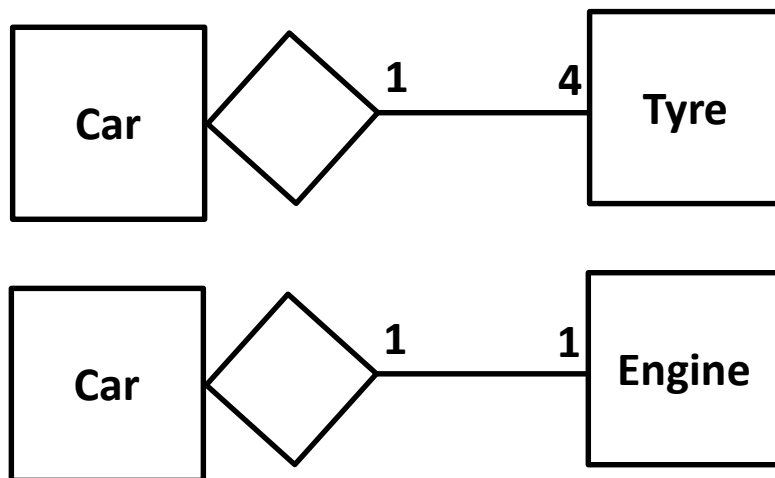
Aggregation vs Composition

Example 1

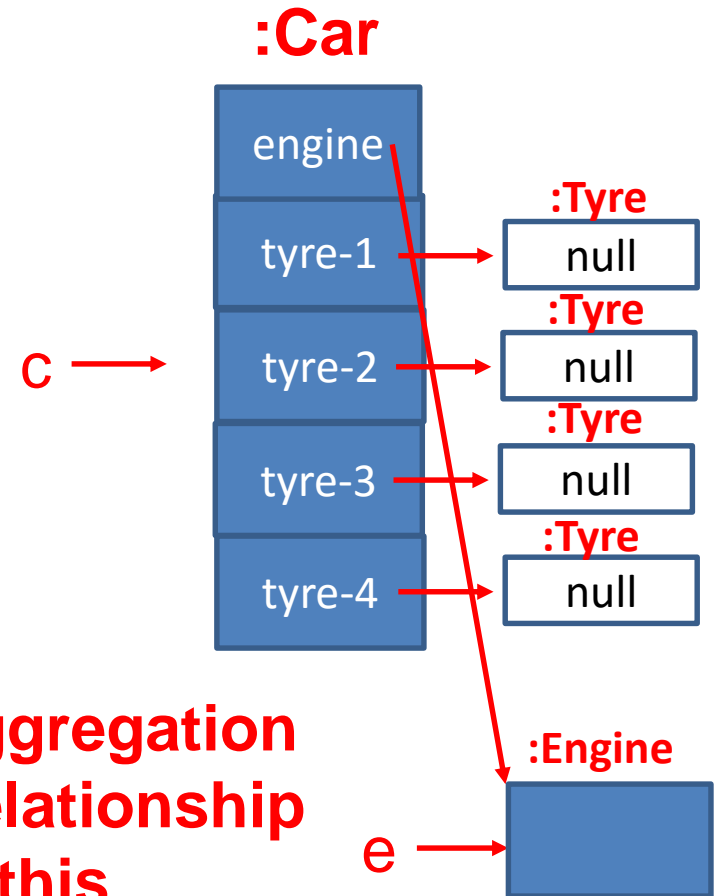


```
Car c = new Car();  
Engine e = new Engine();  
c.setEngine(e);
```

// Similarly You can Set Tyres also



**Aggregation
Relationship
in this
Example**



Aggregation vs Composition

Example 2



- Suppose a parameterized constructor is added in Car class as follows

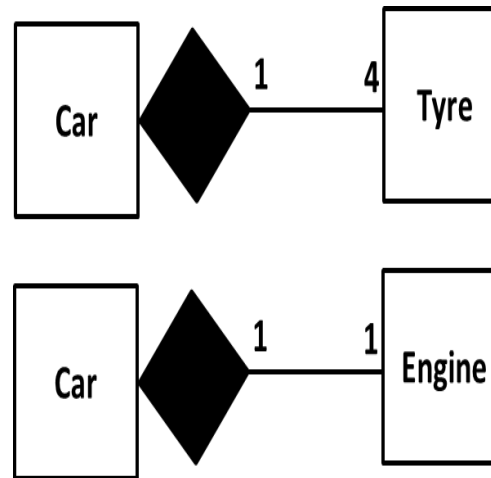
```
class Car
{
```

```
.....
Car(Engine e, Tyre t1, Tyre t2, Tyre t3, Tyre t4)
{
    this.engine = e;
    this.tyre-1 = t1;
    this.tyre-2 = t2;
    this.tyre-3 = t3;
    this.tyre-4 = t4;
}
```

```
.....
} // End of Car class
```

Car c = new Car();  **WRONG STATEMENT**

- You cannot construct a car object without the engine and tyre objects



Composition
Relationship
in this
Example

Thank You