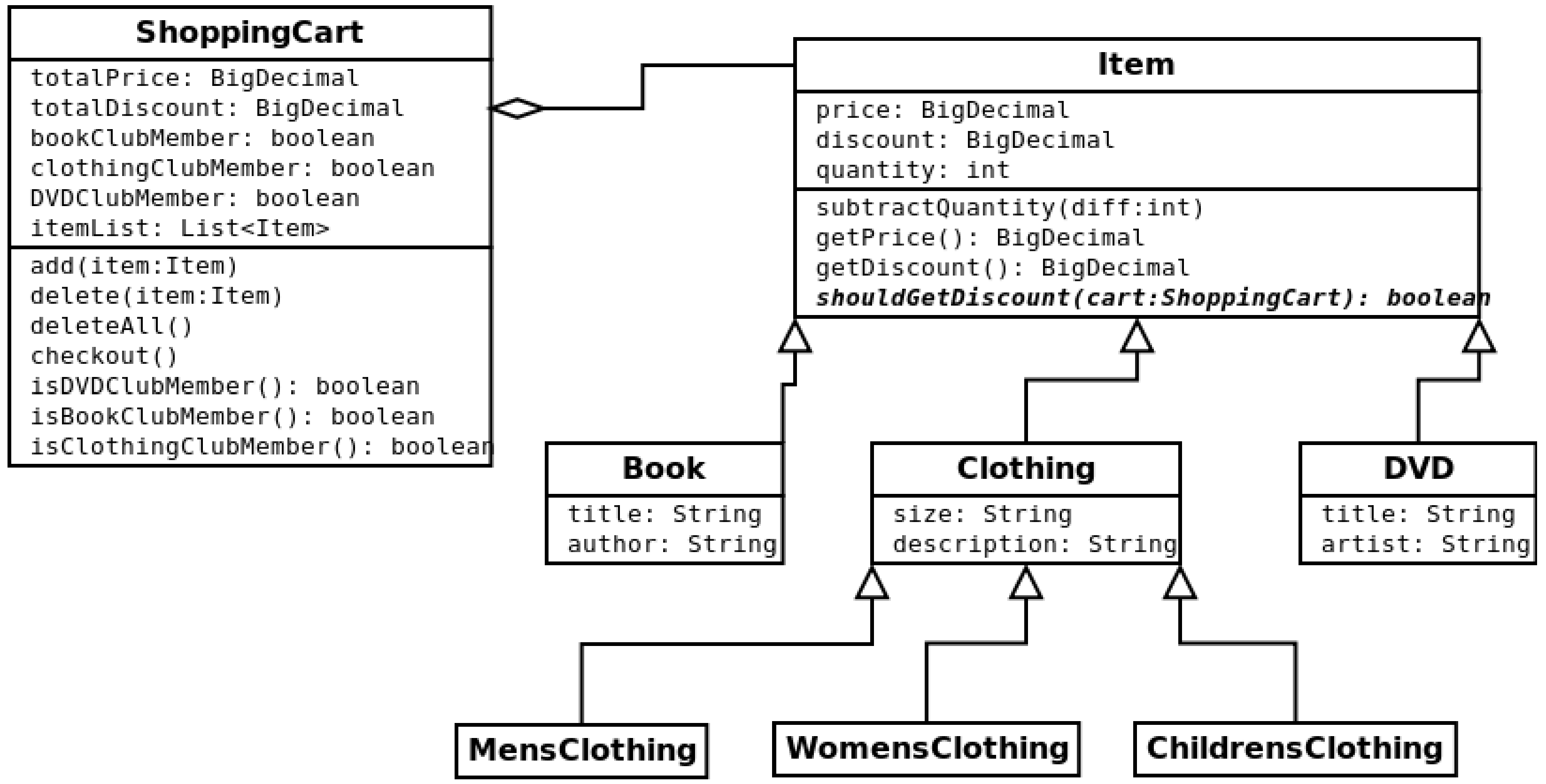
C# OOP (Object-Oriented Programming)

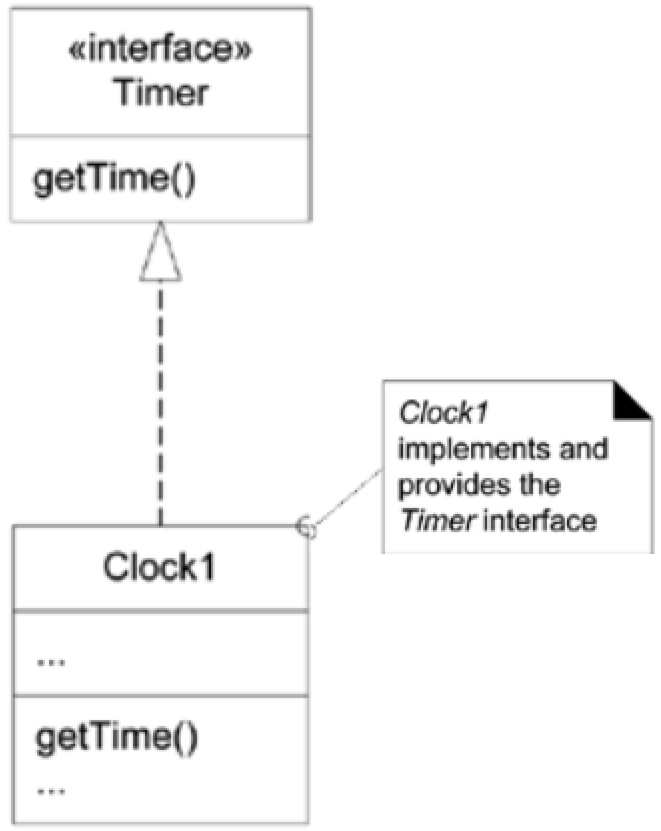
INSTRUCTOR: ADEEN BIN WAHEED

LECTURE 6

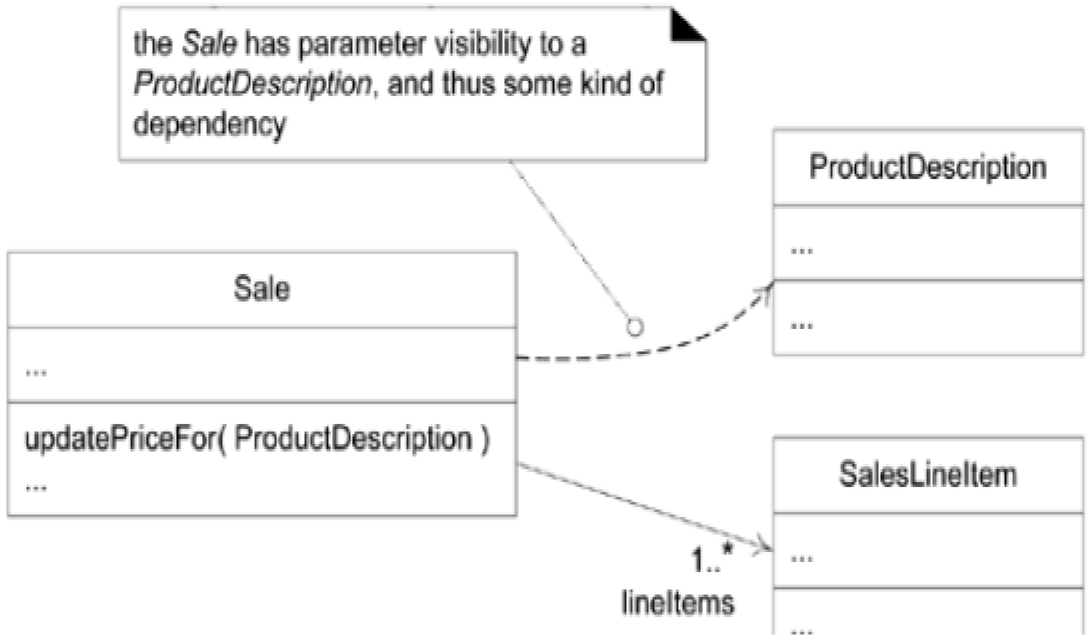
Class Diagram



Realization



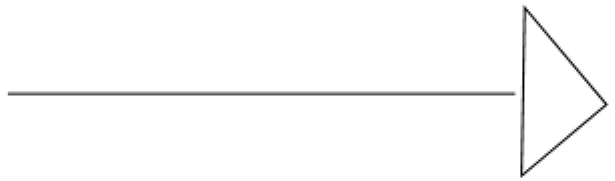
Showing Dependencies


the *Sale* has parameter visibility to a *ProductDescription*, and thus some kind of dependency
ProductDescription
...
...
Sale
...
updatePriceFor(ProductDescription)
...
SalesLineItem
...
...
1..*
lineItems

Generalization

A generalization is a relationship between a general thing(called the super class or parent) and a more specific kind of that thing(called the subclass or child).

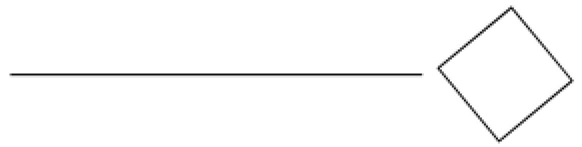
is-a-kind-of relationship.



Aggregation

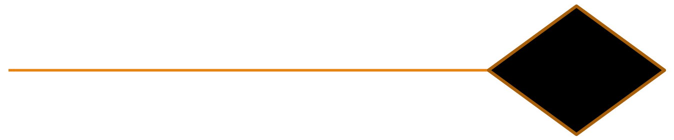
Whole/part relationship

Has-a relationship

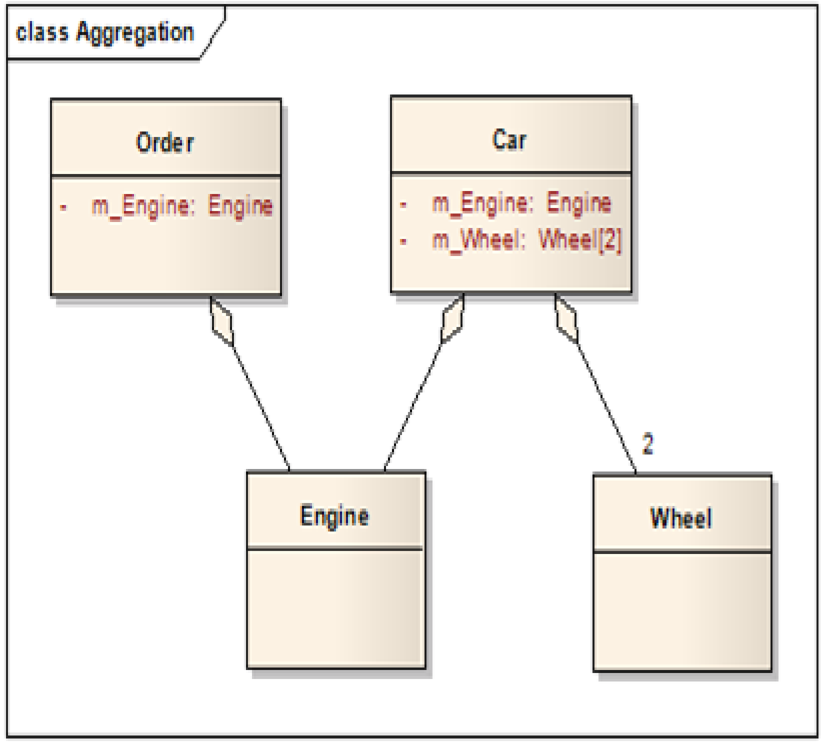


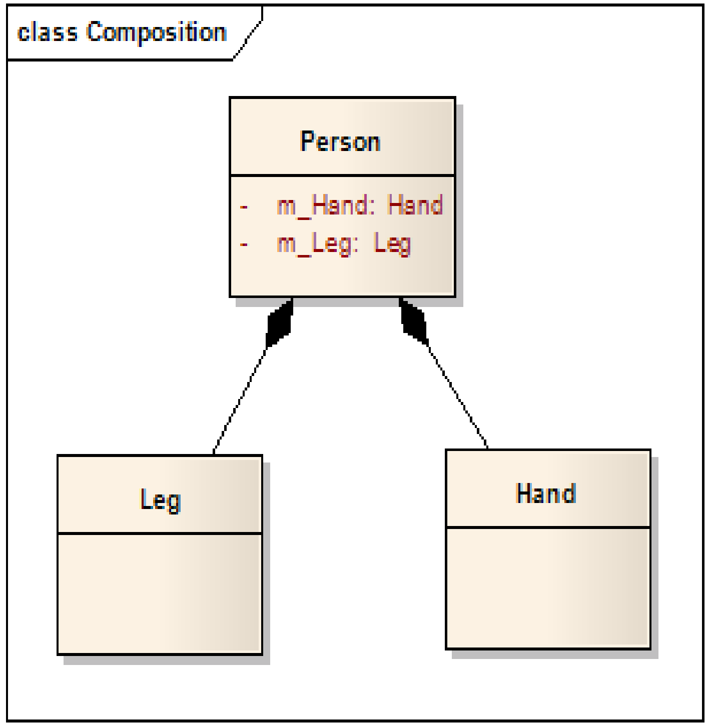
Composition

Has a relationship but there’s a strong life cycle dependency between the two.



Aggregation vs Composition





Some Important Topics

Before jumping towards OOP. Lets us discuss the following topics:

- Interfaces
- Abstract Classes
- Multi-Level Inheritance
- Multiple Inheritance

What is OOP?

Object-oriented programming (OOP) is an engineering approach for building software systems

- Based on the concepts of classes and objects that are used for modeling the real world entities
- Object-oriented programs
- Consist of a group of cooperating objects
- Objects exchange messages, for the purpose of achieving a common objective
- Implemented in object-oriented languages

What OOP’s Claims To Fame?

- Better suited for team development.
- Facilitates utilizing and creating reusable software components.
- Easier GUI programming.
- Easier software maintenance.

- All modern languages are object-oriented: C#, Java, PHP, Perl, C++, ...

Inheritance

A class can inherit another class, inheriting all its data members and methods.

A class can inherit an interface, implementing all the specified methods.

Inheritance “is a” relationship between objects.

In C#, a subclass can inherit only one superclass.

In C#, a sub interface can inherit one super interface

In C#, a class can inherit several interfaces —this is C#’s form of ***multiple inheritance***.

Inheritance

Basic Syntax:

```
<acess-specifier> class <base_class>
{
   ...
}
class <derived_class> : <base_class>
{
   ...
}
```

Inheritance

Running Demo