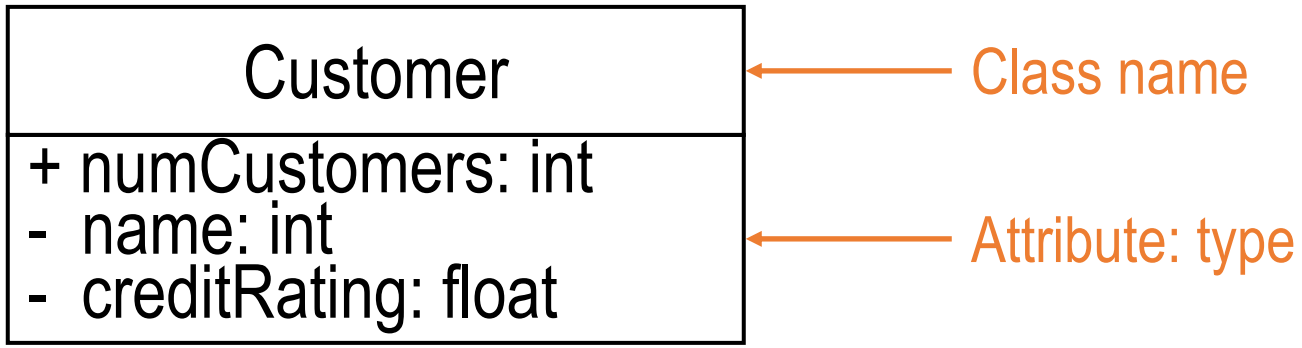
MET CS665 – Software Design and Patterns

UML - Class Diagram

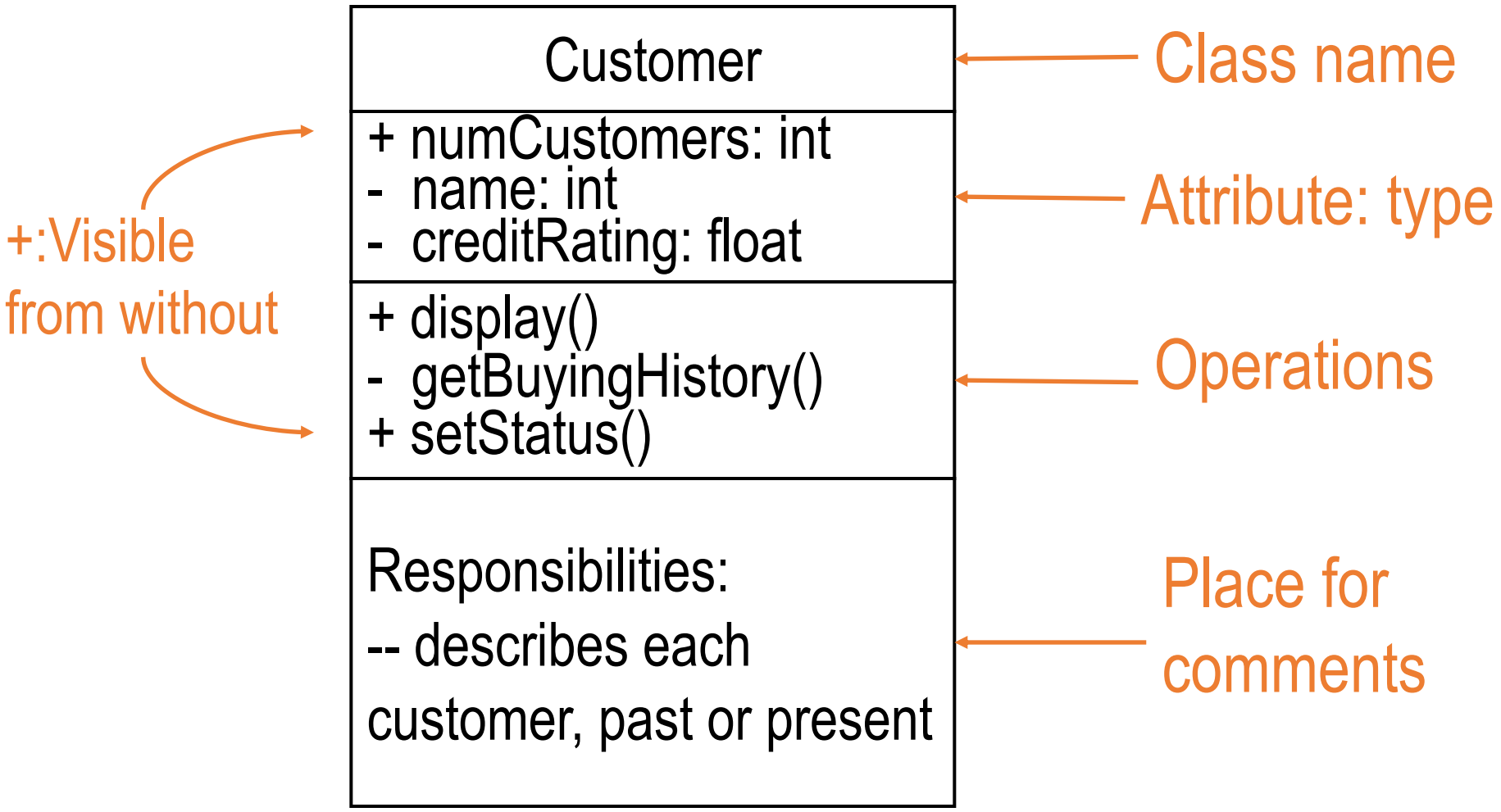
UML Class Diagram

- Class diagram is a structural diagram view that shows:
  - Classes and so the structure of a system
  - Attribute members of classes
  - Relationships between classes

Classes

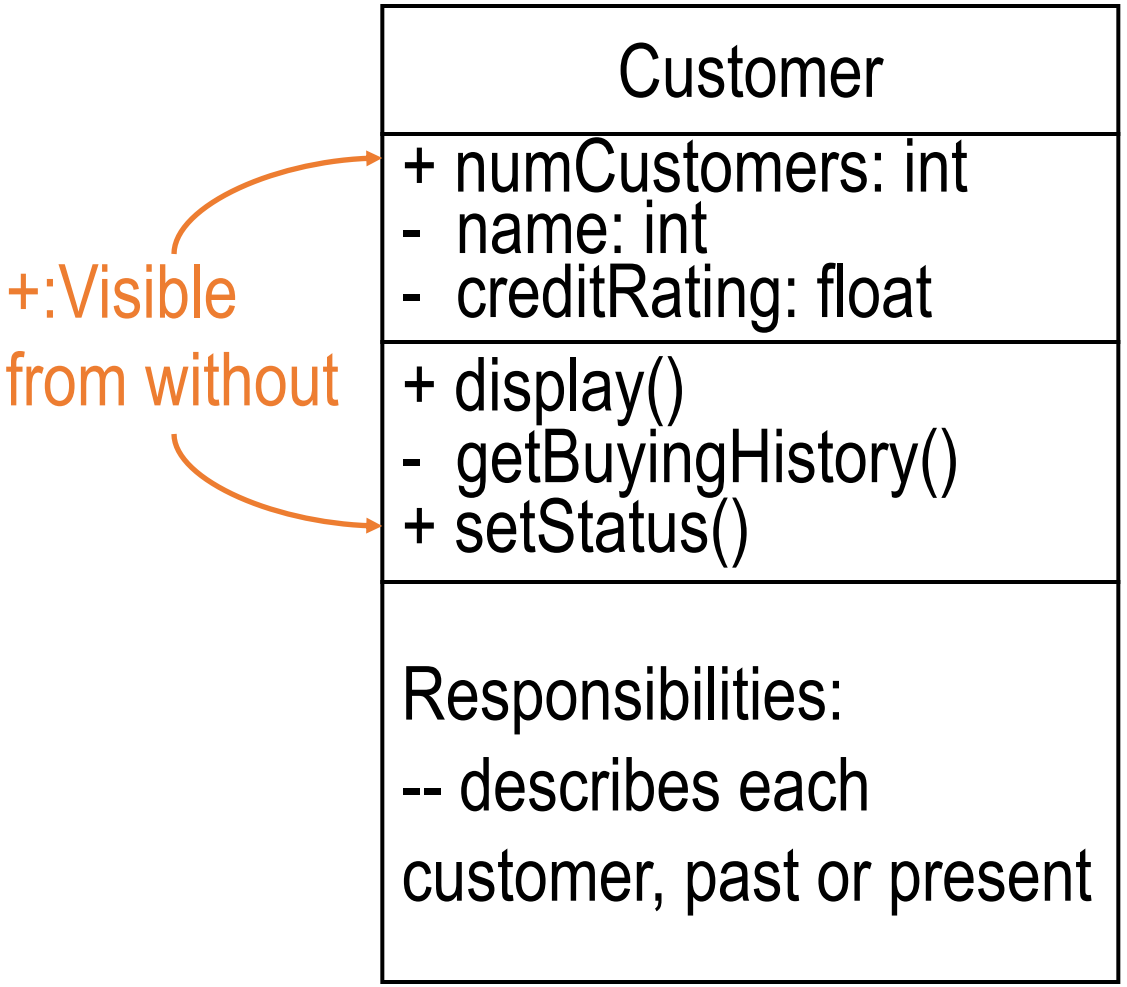

Customer
+ numCustomers: int
- name: int
- creditRating: float
Class name
Attribute: type

Class


Customer
+ numCustomers: int
- name: int
- creditRating: float
+ display()
- getBuyingHistory()
+ setStatus()
Responsibilities:
-- describes each
customer, past or present
+:Visible
from without
Class name
Attribute: type
Operations
Place for
comments

Attribute Visibility

+ Public
# Protected
- Private

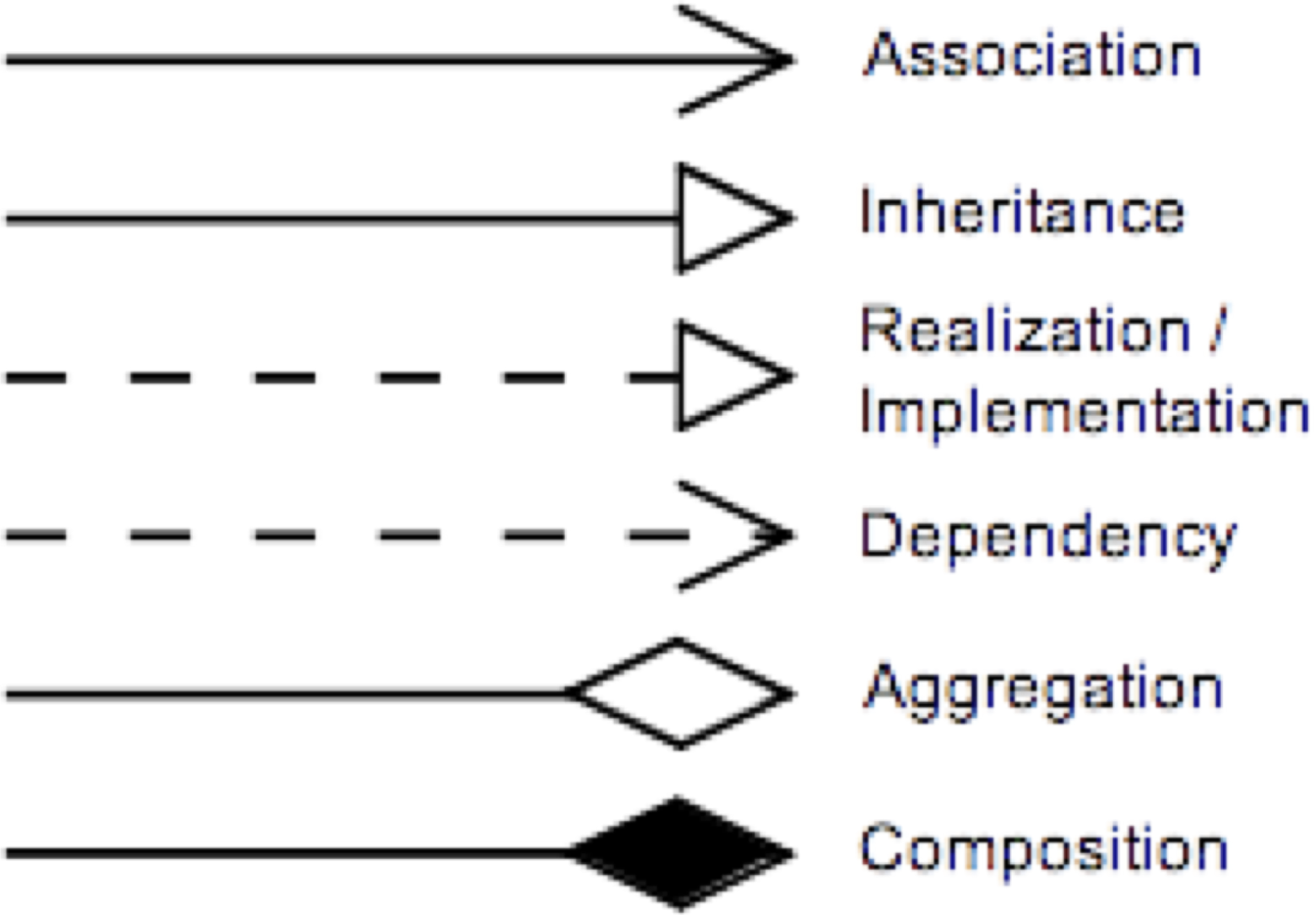


Private - Only the current class will have access to the field or method.

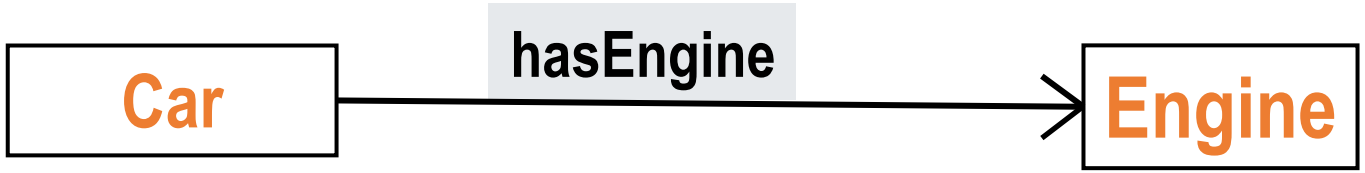
Protected - Only the current class and subclasses (and sometimes also same-package classes) of this class will have access to the field or method.

Public - Any class can refer to the field or call the method.

UML Class Diagram Notations

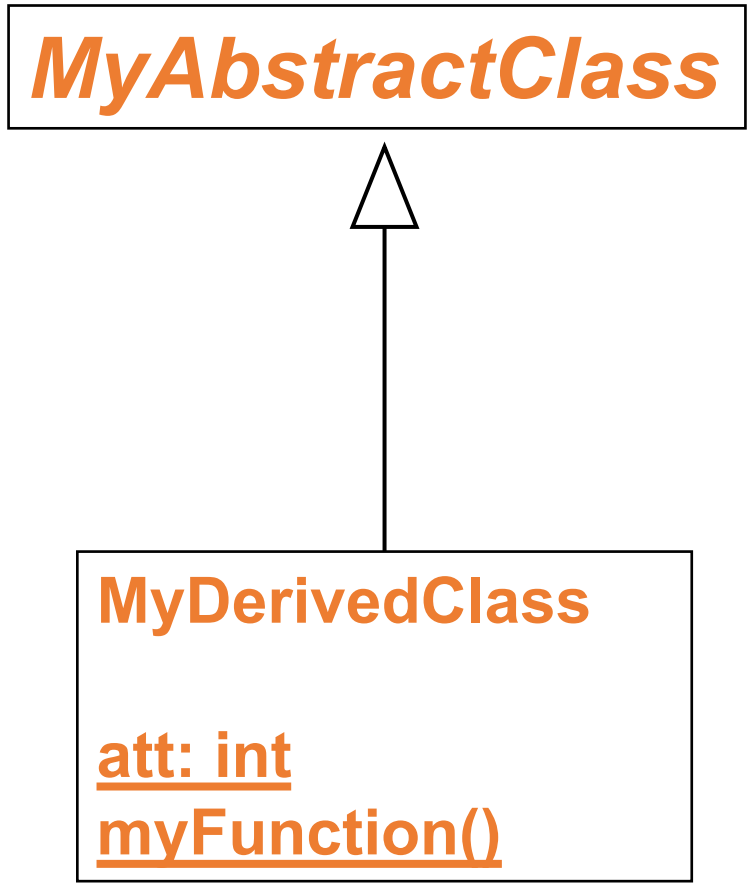


Associations




```
class Car {

    Engine  myCarEngine;
    . . . . .
}

class Engine {

     float hoursePower ;
    . . . . .
}
```

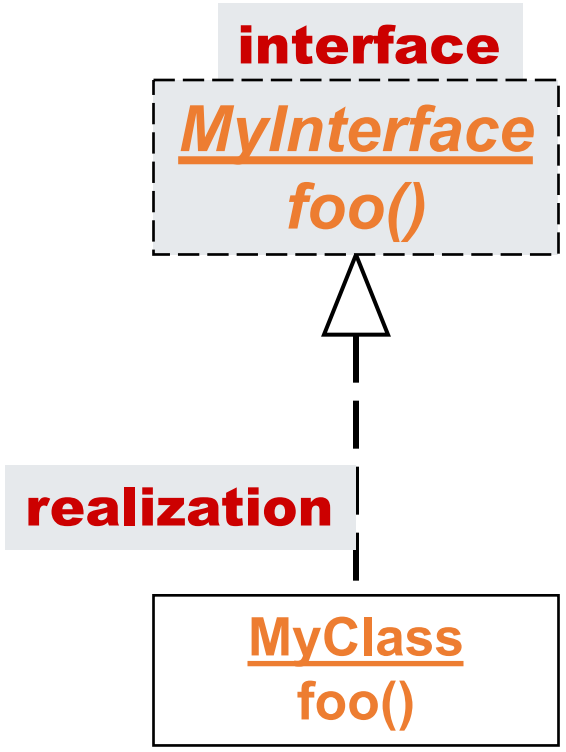
Inheritance




```
package MyPackage;
abstract class MyAbstractClass . . . .

package MyPackage;
class MyDerivedClass extends MyAbstractClass
{
    int att;
    . . . . .
    void myFunction( ReferencedClass r )
    { . .. }
}
```

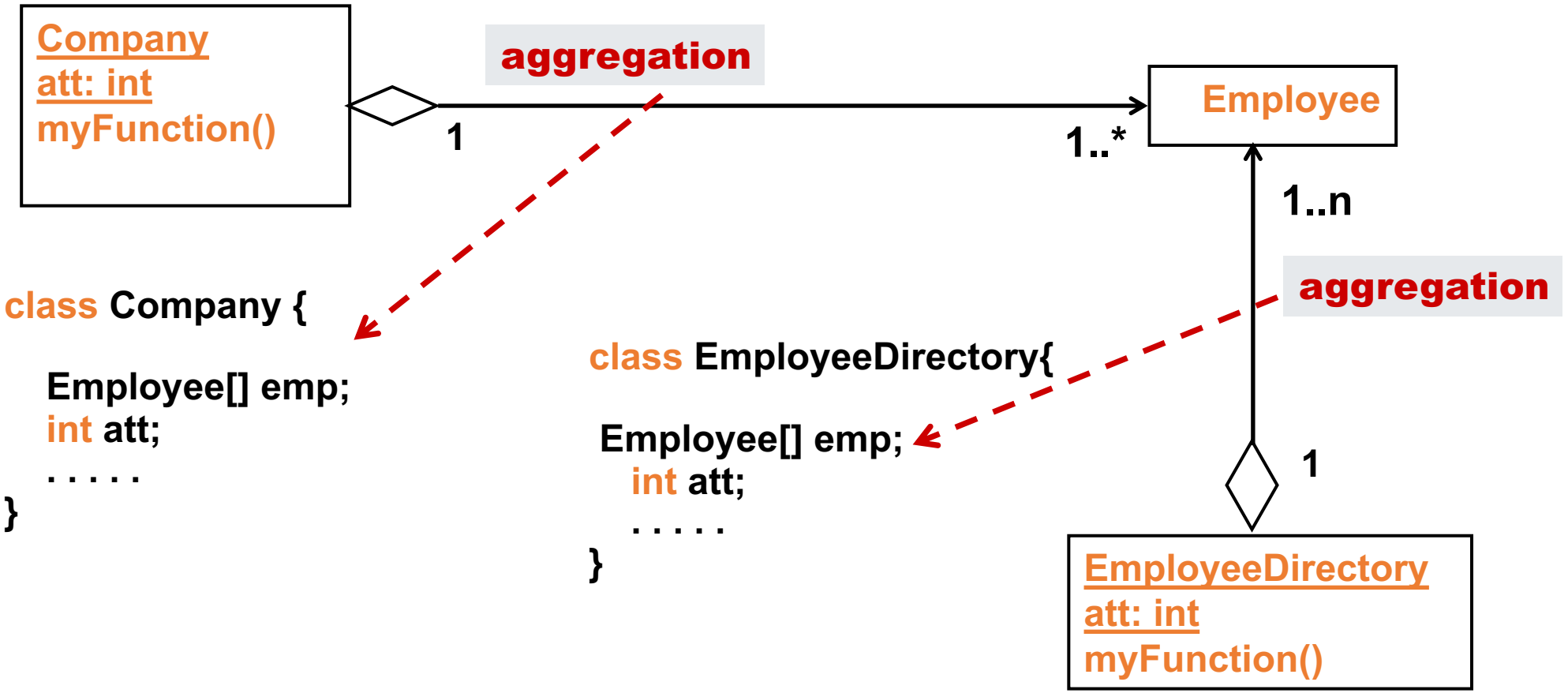
Realization

Realization is where a class satisfies an interface

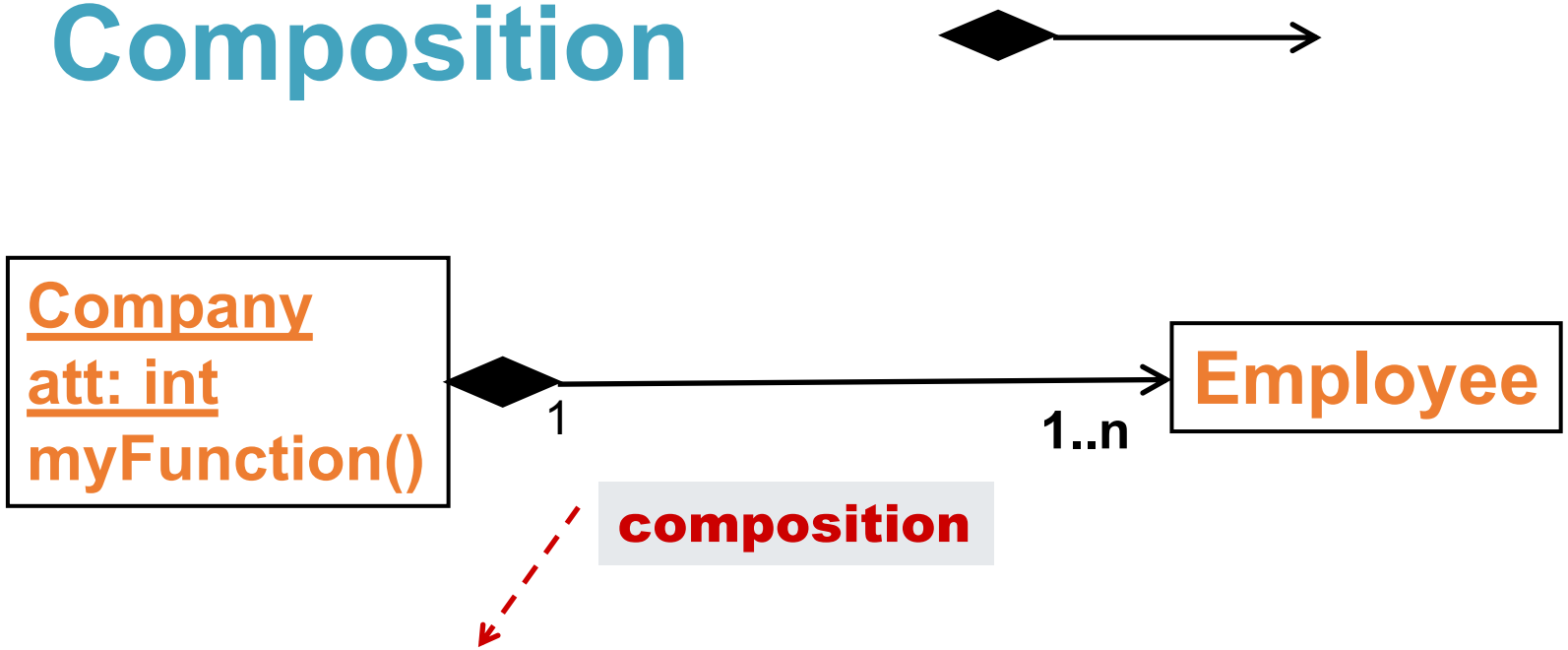



```
interface MyAbstractClass  .  . .


class MyClass implements MyInterface
{
    . . . . .
}
```

Aggregation



Composition

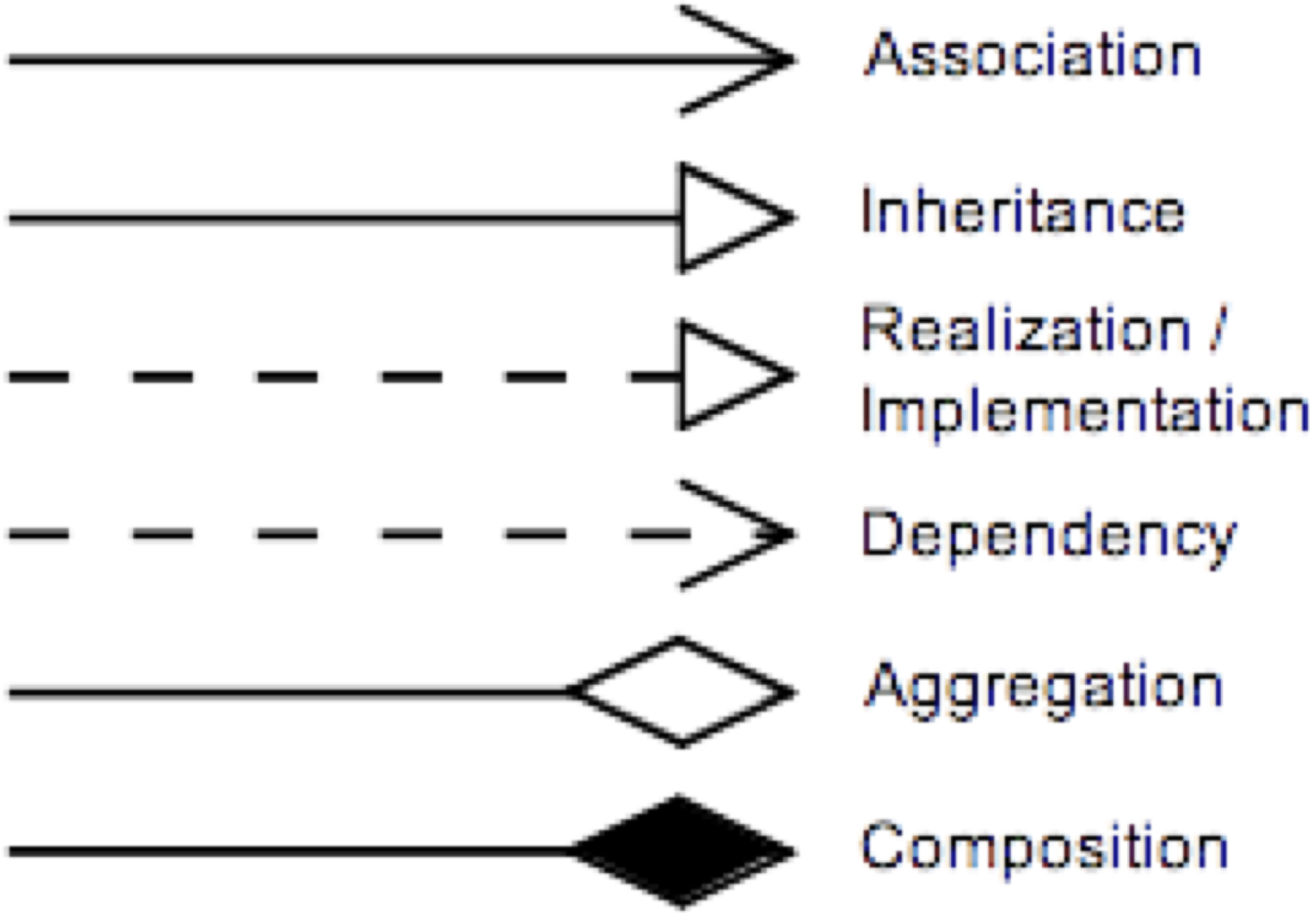



```
class Company {

  class Employee emp;
    { . . . }

. . . . .
}
```

Dependency

A dependency exists between two Classes if changes to the implementation of one Class may cause changes to the other

UML Class Diagram Notations



Customer Mail Application