Lab-Report

Exp. Name	**:**	Association and Aggregation Class Diagram
Course Code	**:**	**CSE-326**
Course Title	**:**	**System Analysis and Design Lab**

Submitted To: Sputa Richard Philip

Senior Lecture

City University

**Department of computer Science and Engineering**

Faculty of Science and Engineering

Submitted By:

ID :171442560

Name :Md. Rasel Miah

Program :CSE(Eve)

**Batch** : 44th

Semester :Spring

Association

The most abstract way to describe static relationship between classes is using the **Association** link, which simply states that there is some kind of a link or a dependency between two classes or more.
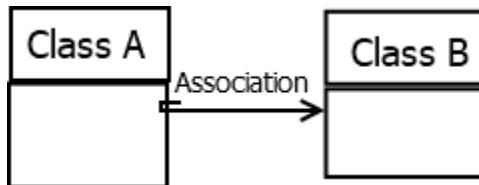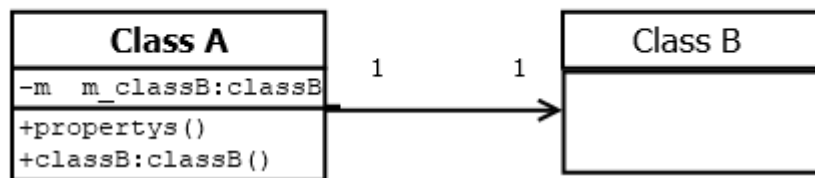

Weak Association

ClassA may be linked to ClassB in order to show that one of its methods includes parameter of ClassB instance, or returns instance of ClassB.
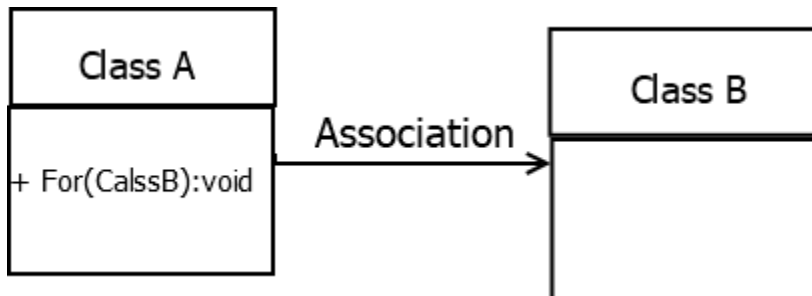
Strong Association

ClassA may also be linked to ClassB in order to show that it holds a reference to ClassB instance.

Class A
-m   m_classB:classB
+propertys()
+classB:classB()

1   1

Class B

Aggregation (Shared Association)

In cases where there's a part-of relationship between ClassA (whole) and ClassB (part), we can be more specific and use the aggregation link instead of the association link, highlighting that the same ClassB instance can also be aggregated by other classes in the application (therefore aggregation is also known as shared association).
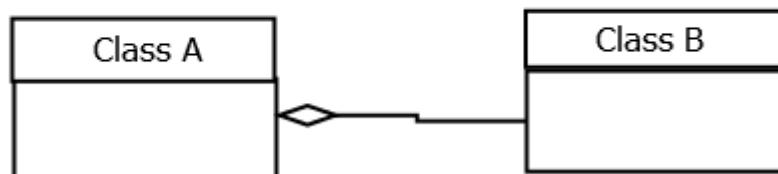
Class A

Class B

It's important to note that the aggregation link **doesn't state** in any way that ClassA owns ClassB **nor** that there's a parent-child relationship (when parent deleted all its child's are being deleted as a result) between the two. Actually, quite the opposite! The aggregation link is usually used to stress the point that ClassA instance is not the exclusive container of ClassB instance, as in fact the same ClassB instance has another container/s.

```
┌─────────────────────┐        ┌──────────────────────┐
│        order        │        │         Car          │
├─────────────────────┤        ├──────────────────────┤
│ -m_Engine:Engine    │        │ -m_Engine:Engine     │
│                     │        │ -m_Wheel:Wheel[2]    │
└─────────────────────┘        └──────────────────────┘
```