**OOPs concepts – What is Association in java?**

In this article we will discuss **Association in Java.** Association establish relationship between two **classes** through their **objects**. The relationship can be one to one, One to many, many to one and many to many.

**Association Example**

class CarClass{

String carName;

double carSpeed;

int carId;

CarClass(String name, double speed, int Id)

{

this.carName=name;

this.carSpeed=speed;

this.carId=Id;

}

}

class Driver{

String driverName;

int driverAge;

Driver(String name, int age){

this.driverName=name;

this.driverAge=age;

}

}

class TransportCompany{

public static void main(String args[])

{

CarClass obj= new CarClass("Ford", 180.15, 9988);

Driver obj2 = new Driver("Andy", 45);

System.out.println(obj2.driverName+" is a driver of car Id: "+obj.carId);

}

}

Output:

Andy is a driver of car Id: 9988

In the above example, there is a one to one relationship(**Association**) between two classes: Car and Driver. Both the classes represents two separate entities.

**Association vs Aggregation vs Composition**

Lets discuss **difference between Association, Aggregation and Composition**:

Although all three are related terms, there are some major differences in the way they relate two classes. **Association** is a relationship between two separate classes which can be of any type say one to one, one to may etc. It joins two entirely separate entities.

[**Aggregation**](http://beginnersbook.com/2013/05/aggregation/) is a special form of association which is a unidirectional one way relationship between classes (or entities), for e.g. Wallet and Money classes. Wallet has Money but money doesn’t need to have Wallet necessarily so its a one directional relationship. In this relationship both the entries can survive if other one ends. In our example if Wallet class is not present, it does not mean that the Money class cannot exist.

**Composition** is a restricted form of Aggregation in which two entities (or you can say classes) are highly dependent on each other. For e.g. Human and Heart. A human needs heart to live and a heart needs a Human body to survive. In other words when the classes (entities) are dependent on each other and their life span are same (if one dies then another one too) then its a composition. Heart class has no sense if Human class is not present.