Aggregation in Java

If a class have an entity reference, it is known as Aggregation. Aggregation represents HAS-A relationship.

Consider a situation, Employee object contains many informations such as id, name, emailId etc. It contains one more object named address, which contains its own informations such as city, state, country, zipcode etc. as given below.

class Employee{

int id;

String name;

Address address;//Address is a class

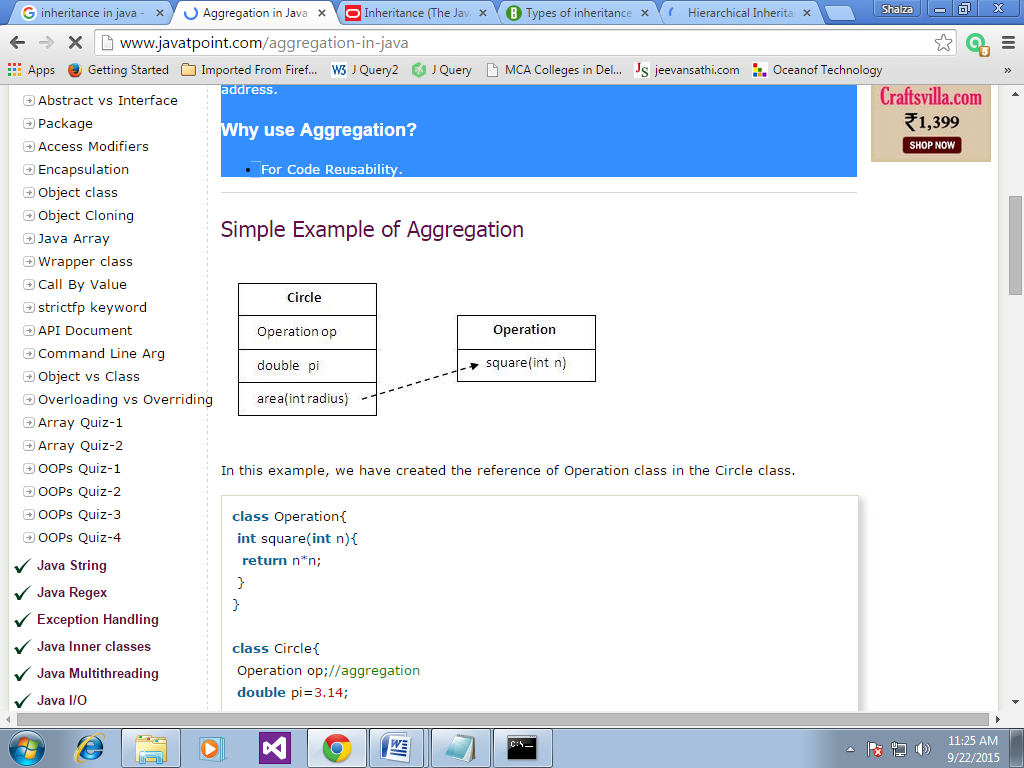
...

}

In such case, Employee has an entity reference address, so relationship is Employee HAS-A address.

**Why use Aggregation?**

For Code Reusability.



**Example:**

class Operation{

int square(int n){

return n\*n; }}

class Circle{

Operation op;//aggregation

double pi=3.14;

double area(int radius){

op=new Operation();

int rsquare=op.square(radius);

//code reusability (i.e. delegates the method call).

return pi\*rsquare;

}

public static void main(String args[]){

Circle c=new Circle();

double result=c.area(5);

System.out.println(result);

}}

Output:78.5

**When use Aggregation?**

1) Code reuse is also best achieved by aggregation when there is no is-a relationship.

2) Inheritance should be used only if the relationship is-a is maintained throughout the lifetime of the objects involved; otherwise, aggregation is the best choice.

**Understanding meaningful example of Aggregation**

**Address.java**

public class Address {

String city,state,country;

public Address(String city, String state, String country) {

this.city = city;

this.state = state;

this.country = country; } }

Make one more class

**Emp.java**

public class Emp {

int id;

String name;

Address address;

public Emp(int id, String name,Address address) {

this.id = id;

this.name = name;

this.address=address; }

void display(){

System.out.println(id+" "+name);

System.out.println(address.city+" "+address.state+" "+address.country); }

public static void main(String[] args) {

Address address1=new Address("gzb","UP","india");

Address address2=new Address("gno","UP","india");

Emp e=new Emp(111,"varun",address1);

Emp e2=new Emp(112,"arun",address2);

e.display();

e2.display(); } }

Output:111 varun

gzb UP india

112 arun

gno UP india

=======================

class Author

{

String authorName;

int age;

String place;

Author(String name,int age,String place)

{

this.authorName=name;

this.age=age;

this.place=place;

}

public String getAuthorName()

{

return authorName;

}

public int getAge()

{

return age;

}

public String getPlace()

{

return place;

}}

class Book

{

String name;

int price;

Author auth;

Book(String n,int p,Author at)

{

this.name=n;

this.price=p;

this.auth=at;

}

public void showDetail()

{

System.out.println("Book is"+name);

System.out.println("price "+price);

System.out.println("Author is "+auth.getAuthorName());

}

}

class Test

{

public static void main(String args[])

{

Author ath=new Author("Me",22,"India");

Book b=new Book("Java",550,ath);

b.showDetail();

}

}

**Output:**

Book is Java.

price is 550.

Author is me.