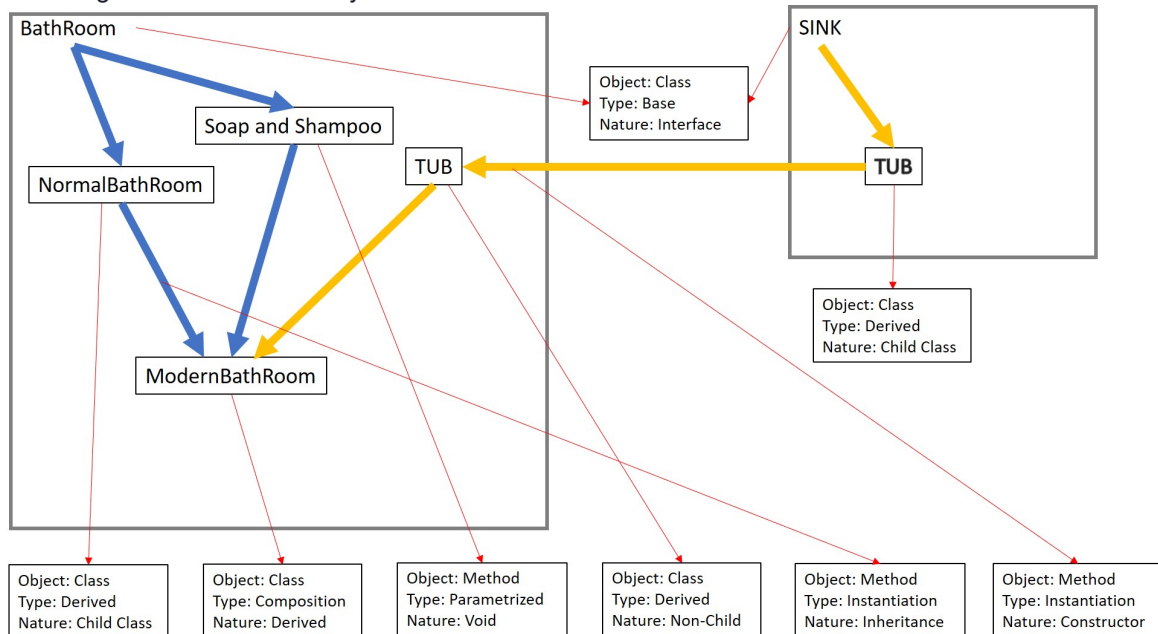


## MD Ismail Hosen

### Kamal Bharakhda

Sorry if this disturbs you.....

Full fledged. let me know if any mistake.



Jul 19, 2021, 12:53 AM

### Kamal Bharakhda

good night brother take care

♥MD Ismail Hosen

Jul 19, 2021, 12:16 AM

### MD Ismail Hosen

Will talk tomorrow

Jul 19, 2021, 12:16 AM

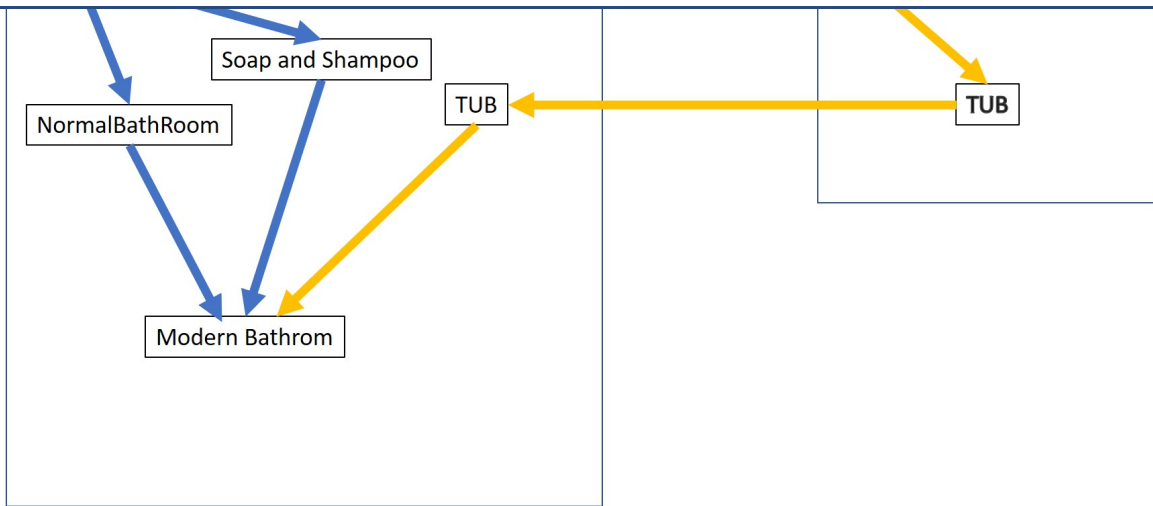
### MD Ismail Hosen

Good night

Jul 19, 2021, 12:15 AM

### Kamal Bharakhda

this will clear everything



♥MD Ismail Hosen

Jul 19, 2021, 12:13 AM

**MD Ismail Hosen**

---

Thanks for clarification and discussion

👍Kamal Bharakhda

Jul 19, 2021, 12:09 AM

**MD Ismail Hosen**

---

Understood

Jul 19, 2021, 12:09 AM

**Kamal Bharakhda**

---

and as you said, we cannot inherit from two class

so first class has to be parent class so in this case present class or derived class will be child class but newly created instance variable or object in side of methods, would be a non-child classes

👍MD Ismail Hosen

Jul 19, 2021, 12:04 AM

**MD Ismail Hosen**

---

Okay

Jul 19, 2021, 12:03 AM

**Kamal Bharakhda**

---

in perspective of inheritated class, newly used functionality in derived class is considered as non-child because he is not a child of base class we have imported it make it better

Jul 19, 2021, 12:02 AM

**MD Ismail Hosen**

---

---

**Kamal Bharakhda**

in simple terms,

what is child class, who have parents. as you said.

we only use methods of parent class in derived class

Now in derived class if due to requirements, we needed a functionality of other class which is tub

then we can say that, tub is not a child of bathroom class but we have used it to create modern class.>

Jul 19, 2021, 12:01 AM

---

**MD Ismail Hosen**

So you are creating objects of another class in another class constructor?

Jul 18, 2021, 11:58 PM

---

**Kamal Bharakhda**

non-child is referred to a class that has been instantiated using constructor method.

what is child class? the class which is derived from the base class

but when we use constructor method to create object (instance variable) we can create it from the mixed of many classes like composition

Jul 18, 2021, 11:57 PM

---

**MD Ismail Hosen**

Although I don't understand non child case yet

Jul 18, 2021, 11:52 PM

---

**MD Ismail Hosen**

This was a very good discussion

Jul 18, 2021, 11:52 PM

---

**Kamal Bharakhda**

:D

Jul 18, 2021, 11:49 PM

---

**Kamal Bharakhda**

the composition, inheritance and interface is now cleared



Case closed

♥MD Ismail Hosen

Jul 18, 2021, 11:49 PM

**MD Ismail Hosen**

---

Yes

Jul 18, 2021, 11:47 PM

**MD Ismail Hosen**

---

But not possible to inherit from multiple class

Jul 18, 2021, 11:47 PM

**Kamal Bharakhda**

---

like derived class is using two interfaces?

Jul 18, 2021, 11:47 PM

**Kamal Bharakhda**

---

Yes!

Jul 18, 2021, 11:47 PM

**MD Ismail Hosen**

---

You can implement more than one interface

Jul 18, 2021, 11:46 PM

**MD Ismail Hosen**

---

That's why code against interface

Jul 18, 2021, 11:46 PM

**MD Ismail Hosen**

---

That's why composition is better than inheritance

Jul 18, 2021, 11:46 PM

**MD Ismail Hosen**

---

Or we can create tub in a general module and set that for bathroom class tub

Jul 18, 2021, 11:45 PM

**MD Ismail Hosen**

---

---

**Kamal Bharakhda**

---

then if tub is his uncle then there would be his external support

Jul 18, 2021, 11:45 PM

---

**MD Ismail Hosen**

---

One children can't have more than one parent

Jul 18, 2021, 11:44 PM

---

**Kamal Bharakhda**

---

I call

TYPE obj = New TYPE as constructor method

Jul 18, 2021, 11:43 PM

---

**MD Ismail Hosen**

---

It's like a real parent child relationship

Jul 18, 2021, 11:43 PM

---

**MD Ismail Hosen**

---

No

Jul 18, 2021, 11:43 PM

---

**Kamal Bharakhda**

---

can we inheritance from two class ?

Jul 18, 2021, 11:43 PM

---

**Kamal Bharakhda**

---

to modifying it.

Jul 18, 2021, 11:43 PM

---

**Kamal Bharakhda**

---

like we have to use constructor methods

Jul 18, 2021, 11:43 PM

---

**Kamal Bharakhda**

---

---

**MD Ismail Hosen**

---

Tub yourTub;

Jul 18, 2021, 11:42 PM

---

**MD Ismail Hosen**

---

In bathroom class

Jul 18, 2021, 11:42 PM

---

**MD Ismail Hosen**

---

Same

Jul 18, 2021, 11:42 PM

---

**MD Ismail Hosen**

---

and not a better design

Jul 18, 2021, 11:42 PM

---

**MD Ismail Hosen**

---

In that case code will be in high coupling

Jul 18, 2021, 11:41 PM

---

**MD Ismail Hosen**

---

Without having bathroom and tub

Jul 18, 2021, 11:41 PM

---

**MD Ismail Hosen**

---

And merge both bathroom and tub

Jul 18, 2021, 11:41 PM

---

**Kamal Bharakhda**

---

how composition class looks in java syntaxially?

Jul 18, 2021, 11:41 PM

---

**MD Ismail Hosen**

---

Like we can have only one class of modern bathroom

---

**MD Ismail Hosen**

---

For better and flexible design yes

Jul 18, 2021, 11:40 PM

---

**Kamal Bharakhda**

---

Yes so when dependency occurs when base class cannot full filling it, we have to go for composition

Jul 18, 2021, 11:40 PM

---

**MD Ismail Hosen**

---

To function properly modern bathroom need tub

Jul 18, 2021, 11:38 PM

---

**Kamal Bharakhda**

---

but to modify it with other facility we have to import some more functionalities which is SINK Class's TUB CLASS which makes it compositon for the developer

Jul 18, 2021, 11:38 PM

---

**MD Ismail Hosen**

---

Modern bathroom has tub dependency

Jul 18, 2021, 11:38 PM

---

**Kamal Bharakhda**

---

Yes so normal bathroom would be child classs of ROOM class

Jul 18, 2021, 11:37 PM

---

**MD Ismail Hosen**

---

It's also called dependency

Jul 18, 2021, 11:37 PM

---

**MD Ismail Hosen**

---

Modern one has it

Jul 18, 2021, 11:37 PM

---

**Kamal Bharakhda**

---

Yes!



---

**MD Ismail Hosen**

Normal bathroom doesn't have tub

Jul 18, 2021, 11:37 PM

---

**Kamal Bharakhda**

because all those are kind of SINK only.... in washbasin we are washing hands, in kitchen sink we have washing plates, and in bathrom tub we are washing our self

Jul 18, 2021, 11:36 PM

---

**Kamal Bharakhda**

because if we create SINK base class then we can now implement TUB and WASHBASIN, KITCHEN SINK child classes

Jul 18, 2021, 11:35 PM

---

**MD Ismail Hosen**

But tub is independent

Jul 18, 2021, 11:35 PM

---

**MD Ismail Hosen**

We will use tub from bathroom

Jul 18, 2021, 11:35 PM

---

**MD Ismail Hosen**

Yes

Jul 18, 2021, 11:35 PM

---

**Kamal Bharakhda**

but tub class should be independent

Jul 18, 2021, 11:35 PM

---

**MD Ismail Hosen**

So if we use from bathroom then composition

Jul 18, 2021, 11:35 PM

---

**MD Ismail Hosen**

We use tub class from bathroom or independently



---

**Kamal Bharakhda**

---

ohhh I now got what you are thinking of child and non-child

Jul 18, 2021, 11:34 PM

---

**MD Ismail Hosen**

---

Tub has relationships with bathroom but not as a parent child one

Jul 18, 2021, 11:33 PM

---

**Kamal Bharakhda**

---

see if we don't have ROOM class in our project... then Bathroom has to be independent class

Jul 18, 2021, 11:33 PM

---

**MD Ismail Hosen**

---

Both bathroom and tub is not a base class

Jul 18, 2021, 11:33 PM

---

**MD Ismail Hosen**

---

Base class should have child

Jul 18, 2021, 11:33 PM

---

**MD Ismail Hosen**

---

But how bathroom or tub is base class?

Jul 18, 2021, 11:32 PM

---

**Kamal Bharakhda**

---

no no bro...I'm not telling you...tell me if I'm right or not

Jul 18, 2021, 11:32 PM

---

**MD Ismail Hosen**

---

I understand it

Jul 18, 2021, 11:32 PM

---

**Kamal Bharakhda**

---

NormalBathroom (Child Class) + Tub = ModernBathroom

+ means composition

---

**Kamal Bharakhda**

we can create ModernBathroom Class by including Tub in NormalBathRoom Class

Jul 18, 2021, 11:31 PM

---

**MD Ismail Hosen**

Hmm

Jul 18, 2021, 11:30 PM

---

**Kamal Bharakhda**

so I guess, bathroom is child class  
while tub is independent class which has nothing to do with ROOM

Jul 18, 2021, 11:30 PM

---

**Kamal Bharakhda**

a flat or home is made many rooms or different kind of rooms

which is same analogy of tub inside bathroom

Jul 18, 2021, 11:29 PM

---

**MD Ismail Hosen**

But how bathroom or tub is?

Jul 18, 2021, 11:28 PM

---

**MD Ismail Hosen**

Room is a base class

Jul 18, 2021, 11:28 PM

---

**Kamal Bharakhda**

have floor but with different property like floor color

Jul 18, 2021, 11:28 PM

---

**Kamal Bharakhda**

have window to fill room with fresh air

Jul 18, 2021, 11:27 PM

---

**Kamal Bharakhda**

to start fan

---

**Kamal Bharakhda**

---

with common methods like all kind of rooms have switch to ligh bulb

Jul 18, 2021, 11:27 PM

---

**Kamal Bharakhda**

---

in that case bathroom would be child class of ROOM Class

Jul 18, 2021, 11:26 PM

---

**Kamal Bharakhda**

---

and bathroom too

Jul 18, 2021, 11:26 PM

---

**Kamal Bharakhda**

---

but now, bathroom is also base class as we do not have TYPE of bathroom in our home. we can have different types of ROOMS in our home

like ROOM CLAAS is base class  
child class would be hall, kitchen, bedroom

Jul 18, 2021, 11:26 PM

---

**Kamal Bharakhda**

---

if we only want tub in use, if want some facility of bathroom while bathing in tub, then it will be composition classs

Jul 18, 2021, 11:25 PM

---

**MD Ismail Hosen**

---

And bathroom?

Jul 18, 2021, 11:24 PM

---

**Kamal Bharakhda**

---

tub would be a base class not child nor non-child

Jul 18, 2021, 11:24 PM

---

**MD Ismail Hosen**

---

Is tub a non child class?

Jul 18, 2021, 11:23 PM

---

**MD Ismail Hosen**

---



Jul 18, 2021, 11:21 PM

---

**MD Ismail Hosen**

Check images

Jul 18, 2021, 11:21 PM

---

**Kamal Bharakhda**

lol

Jul 18, 2021, 11:20 PM

---

**Kamal Bharakhda**

means irregular shape

Jul 18, 2021, 11:20 PM

---

**Kamal Bharakhda**

what do you mean by amoeba class?

Jul 18, 2021, 11:17 PM

---

**MD Ismail Hosen**

Fixed for other three class

Jul 18, 2021, 11:16 PM

---

**MD Ismail Hosen**

Override in only amoeba class

Jul 18, 2021, 11:15 PM

---

**MD Ismail Hosen**

Good analogy

Jul 18, 2021, 11:15 PM

---

**MD Ismail Hosen**

## So, Brad the OO guy got the chair, right?

*Not so fast.* Larry found a flaw in Brad's approach. And, since he was sure that if he got the chair he'd also get Lucy in accounting, he had to turn this thing around.

**LARRY:** You've got duplicated code! The rotate procedure is in all four Shape things.

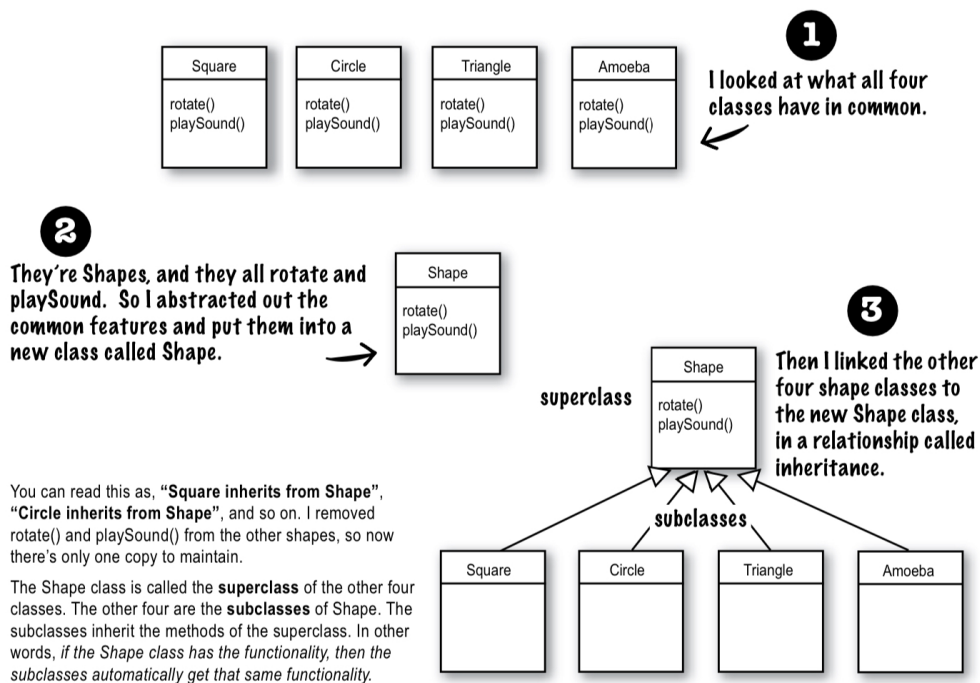
**BRAD:** It's a *method*, not a *procedure*. And they're *classes*, not *things*.

**LARRY:** Whatever. It's a stupid design. You have to maintain *four* different rotate "methods". How can that ever be good?

**BRAD:** Oh, I guess you didn't see the final design. Let me show you how OO **inheritance** works, Larry.



What Larry wanted  
(figured the chair would impress her)



Jul 18, 2021, 11:15 PM

Kamal Bharakhda

future changes....

Jul 18, 2021, 11:14 PM

Kamal Bharakhda

yeah that's another complexity...

Jul 18, 2021, 11:14 PM

MD Ismail Hosen

Kamal Bharakhda

Jul 18, 2021, 11:13 PM

**Kamal Bharakhda**

---

thanks buddy it was a fruitful discussion

Jul 18, 2021, 11:13 PM

**MD Ismail Hosen**

---

Yes

Jul 18, 2021, 11:13 PM

**Kamal Bharakhda**

---

Yes! but there's no point of overriding 9 out of 10 methods of class...so in that case it is better to implement base class as interface

Jul 18, 2021, 11:12 PM

**MD Ismail Hosen**

---

If we have method override capability then inheritance is good option

Jul 18, 2021, 11:12 PM

**MD Ismail Hosen**

---

Close enough

Jul 18, 2021, 11:11 PM

**Kamal Bharakhda**

---

when multiple members are affected then we should go for interface  
when 1 or two members are affected then we should keep into inheritance.

Jul 18, 2021, 11:10 PM

**Kamal Bharakhda**

---

I understand! now, because, if we talk about oven class, we should implement it as for inheritance when derived class type is just same, like Microwave1 and Microwave2 but Microwave2 has only method which is different then 1 so we can override it else everything is same in technical point of view...

we should implement oven class as interface when the technology of both derived class is different for most of the members

Jul 18, 2021, 11:10 PM

**MD Ismail Hosen**

---

---

**MD Ismail Hosen**

Sometimes inheritance is very useful

Jul 18, 2021, 11:07 PM

---

**Kamal Bharakhda**

and that's why they have only gave option of interface in VBA...as inheritance is absolutely no required because in VBA, new objects is created only when it has different logic.

Jul 18, 2021, 11:07 PM

---

**MD Ismail Hosen**

Client own implementation

Jul 18, 2021, 11:06 PM

---

**MD Ismail Hosen**

So client class can do your own implementation

Jul 18, 2021, 11:06 PM

---

**MD Ismail Hosen**

That's why it is best to code against interface

Jul 18, 2021, 11:06 PM

---

**MD Ismail Hosen**

Yes

Jul 18, 2021, 11:05 PM

---

**MD Ismail Hosen**

Exactly

Jul 18, 2021, 11:05 PM

---

**Kamal Bharakhda**

?

Jul 18, 2021, 11:04 PM

---

**Kamal Bharakhda**

in interface, logic is always implements explicitly for all the derived class

Jul 18, 2021, 11:03 PM

---

**Kamal Bharakhda**

hmm means if we have different logic of method then we should go for Interface  
but when the logic is same, then we can go for inheritance and we can also override by  
implementing the same method's logic in deriving class

Jul 18, 2021, 11:01 PM

---

**MD Ismail Hosen**

Interface is useful where method name is same but implementation is different for different class

Jul 18, 2021, 11:00 PM

---

**MD Ismail Hosen**

No need to change (although we can change by overriding)

Jul 18, 2021, 10:59 PM

---

**MD Ismail Hosen**

So same method implementation for all class

Jul 18, 2021, 10:59 PM

---

**MD Ismail Hosen**

When method implementation is not different for different object then we use inheritance

Jul 18, 2021, 10:58 PM

---

**MD Ismail Hosen**

Inheritance is different things

Jul 18, 2021, 10:58 PM

---

**Kamal Bharakhda**

now, you mean to say, if we use interface then we don't have to specify explicitly inheritance ?

Jul 18, 2021, 10:50 PM

---

**Kamal Bharakhda**

instead of word extends

Jul 18, 2021, 10:49 PM



where else in c# we just use colon

Jul 18, 2021, 10:49 PM

**MD Ismail Hosen**

---

Yes

Jul 18, 2021, 10:49 PM

**MD Ismail Hosen**

---

In both induction and microwave

Jul 18, 2021, 10:49 PM

**Kamal Bharakhda**

---

for inheritance we use extends in java

Jul 18, 2021, 10:49 PM

**MD Ismail Hosen**

---

It will be implement IOven

Jul 18, 2021, 10:49 PM

**MD Ismail Hosen**

---

In vba

Jul 18, 2021, 10:49 PM

**Kamal Bharakhda**

---

ok ok

Jul 18, 2021, 10:48 PM

**MD Ismail Hosen**

---

In Java it will be public class Induction implement IOven

Jul 18, 2021, 10:48 PM

**Kamal Bharakhda**

---

yeah please show it to me in Java

Jul 18, 2021, 10:48 PM

**MD Ismail Hosen**

---

---

**Kamal Bharakhda**

now how do you use Interface keyword in derived class?

Jul 18, 2021, 10:47 PM

---

**Kamal Bharakhda**

see

```
public class IOven {  
    public void ignite() {  
    }  
}
```

```
public class Induction : IOven {  
    public void ignite() {  
    }  
}
```

👍 MD Ismail Hosen

Jul 18, 2021, 10:46 PM

---

**MD Ismail Hosen**

And implement them in induction and microwave class

Jul 18, 2021, 10:44 PM

---

**MD Ismail Hosen**

So declare that ignite method in IOven class

Jul 18, 2021, 10:44 PM

---

**Kamal Bharakhda**

yup

Jul 18, 2021, 10:44 PM

---

**MD Ismail Hosen**

Jul 18, 2021, 10:43 PM

---

**MD Ismail Hosen**

So induction and microwave has that method

Jul 18, 2021, 10:43 PM

---

**MD Ismail Hosen**

Let's say we have a method call ignite

Jul 18, 2021, 10:43 PM

---

**MD Ismail Hosen**

Method of IOven class

Jul 18, 2021, 10:43 PM

---

**MD Ismail Hosen**

We will implement method in induction class

Jul 18, 2021, 10:43 PM

---

**Kamal Bharakhda**

how interface is used along with impliments

Jul 18, 2021, 10:41 PM

---

**MD Ismail Hosen**

And will get polymorphism

Jul 18, 2021, 10:40 PM

---

**Kamal Bharakhda**

IOwen MyOwen = New Microwave

👍 MD Ismail Hosen

Jul 18, 2021, 10:39 PM

---

**MD Ismail Hosen**

Instead of concrete class

Jul 18, 2021, 10:39 PM

---

**Kamal Bharakhda**

Jul 18, 2021, 10:39 PM

**MD Ismail Hosen**

---

Code against interface

Jul 18, 2021, 10:39 PM

**Kamal Bharakhda**

---

Yes!

Jul 18, 2021, 10:39 PM

**MD Ismail Hosen**

---

Later if we will get another type then that will be AnotherOven

Jul 18, 2021, 10:39 PM

**Kamal Bharakhda**

---

agreed

Jul 18, 2021, 10:38 PM

**Kamal Bharakhda**

---

IOven would be interface

Jul 18, 2021, 10:38 PM

**MD Ismail Hosen**

---

IOven and induction,Microwave

Jul 18, 2021, 10:38 PM

**Kamal Bharakhda**

---

I hope we have made this conversation in group

Jul 18, 2021, 10:37 PM

**MD Ismail Hosen**

---

Or we can use interface here

Jul 18, 2021, 10:37 PM

**Kamal Bharakhda**

---

understood

---

**Kamal Bharakhda**

yes! bro...

Jul 18, 2021, 10:37 PM

---

**MD Ismail Hosen**

And inherit them in induction and microwave class

Jul 18, 2021, 10:36 PM

---

**MD Ismail Hosen**

Just add common method and instance variable in oven class

Jul 18, 2021, 10:36 PM

---

**MD Ismail Hosen**

Exactly

Jul 18, 2021, 10:36 PM

---

**MD Ismail Hosen**

Yes

Jul 18, 2021, 10:36 PM

---

**Kamal Bharakhda**

so induction IS child of oven

Jul 18, 2021, 10:36 PM

---

**Kamal Bharakhda**

well, oven could be the super class when they have two type of owne induction and microwave

👍 MD Ismail Hosen

Jul 18, 2021, 10:35 PM

---

**MD Ismail Hosen**

Bathroom HAS tub, Kitchen HAS oven and so on

Jul 18, 2021, 10:35 PM

---

**Kamal Bharakhda**

yup!

Jul 18, 2021, 10:34 PM

---

This IS and HAS test is very good test to understand them

Jul 18, 2021, 10:34 PM

---

**MD Ismail Hosen**

---

It's very good book and easy to understand

Jul 18, 2021, 10:33 PM

---

**Kamal Bharakhda**

---

i fianlly understood the difference between inheritance and composition

👍 MD Ismail Hosen

Jul 18, 2021, 10:33 PM

---

**Kamal Bharakhda**

---

thanks buddy

Jul 18, 2021, 10:33 PM

---

**Kamal Bharakhda**

---

this is absolutely brilliant analogy

Jul 18, 2021, 10:33 PM

---

**MD Ismail Hosen**

---

Yes

Jul 18, 2021, 10:33 PM

---

**MD Ismail Hosen**

---

No need to understand code of this book. Just read explanation and see those pictures

Jul 18, 2021, 10:33 PM

---

**Kamal Bharakhda**

---

but owen is definately not an kitchen at alll

Jul 18, 2021, 10:33 PM

---

**MD Ismail Hosen**

---

Exactly

Jul 18, 2021, 10:32 PM

am I correct with above example

Jul 18, 2021, 10:29 PM

### **Kamal Bharakhda**

---

like....

kitchen and owen

kitchend has owen which is composition class...we can use owen through kitchet class as well as directly by owen class.

Jul 18, 2021, 10:28 PM

### **MD Ismail Hosen**

---

objects have **state** and **behavior**

106 of 722

## Remember: a class describes what an object knows and what an object does

A **class** is the **blueprint** for an object. When you write a class, you're describing how the JVM should make an object of that type. You already know that every object of that type can have different *instance variable* values. But what about the methods?

### Can every object of that type have different method behavior?

Well... *sort of*.\*

Every instance of a particular class has the same methods, but the methods can *behave* differently based on the value of the instance variables.

The Song class has two instance variables, *title* and *artist*. The play() method plays a song, but the instance you call play() on will play the song represented by the value of the *title* instance variable for that instance. So, if you call the play() method on one instance you'll hear the song "Politik", while another instance plays "Darkstar". The method code, however, is the same.

```
void play() {  
    soundPlayer.playSound(title);  
}
```

```
Song t2 = new Song();  
t2.setArtist("Travis");  
t2.setTitle("Sing");  
Song s3 = new Song();  
s3.setArtist("Sex Pistols");  
s3.setTitle("My Way");
```

\*Yes, another stunningly clear answer!

72 chapter 4

**instance variables**  
(state)

**methods**  
(behavior)

Song	
title	
artist	
setTitle()	
setArtist()	
play()	

**knows**

**does**

five instances  
of class Song

Politik  
Coldplay

My Way  
Sinatra

Darkstar  
Grateful Dead

Sing  
Travis

My Way  
Sex Pistols



Song

t2.play();

Calling play() on this instance  
will cause "Sing" to play.



Song

s3.play();

Calling play() on this instance  
will cause "My Way" to play.  
(but not the Sinatra one)



Jul 18, 2021, 10:28 PM

Kamal Bharakhda

yes yes this is definetly composition

Jul 18, 2021, 10:27 PM

Kamal Bharakhda

got it .... bathroom has used tub class to enriching it's own self

Jul 18, 2021, 10:26 PM

Kamal Bharakhda



---

**MD Ismail Hosen**

---

Bathroom can use method of tub(is fill with water)

Jul 18, 2021, 10:26 PM

---

**MD Ismail Hosen**

---

No. Reverse

Jul 18, 2021, 10:26 PM

---

**MD Ismail Hosen**

---

One class hold object of another class

Jul 18, 2021, 10:25 PM

---

**Kamal Bharakhda**

---

mean, tub can use the method like, tap, shower of bathroom

Jul 18, 2021, 10:25 PM

---

**MD Ismail Hosen**

---

That is called composition

Jul 18, 2021, 10:25 PM

---

**Kamal Bharakhda**

---

okay you mean to say! IS-A approach is child class and HAS-A approach is non-child class.

Jul 18, 2021, 10:25 PM

---

**MD Ismail Hosen**

---

But in bathroom class it has an instance variable of tub class

Jul 18, 2021, 10:23 PM

---

**MD Ismail Hosen**

---

Bathroom also object

Jul 18, 2021, 10:23 PM

---

**MD Ismail Hosen**

---

Tub is object

---

**MD Ismail Hosen**

---

<https://www.pdfdrive.com/head-first-java-e175339683.html>  
<https://www.pdfdrive.com/head-first-java-e175339683.html>

Jul 18, 2021, 10:18 PM

---

**MD Ismail Hosen**

---

You will find it in pdf drive

Jul 18, 2021, 10:16 PM

---

**MD Ismail Hosen**

---

Book name : Head first java

Jul 18, 2021, 10:16 PM

---

**Kamal Bharakhda**

---

:)

Jul 18, 2021, 10:15 PM

---

**Kamal Bharakhda**

---

yes please share the link

Jul 18, 2021, 10:15 PM

---

**MD Ismail Hosen**

---

Can you download it? It's 47 Mb in size. And in here net connection is bad

Jul 18, 2021, 10:15 PM

---

**MD Ismail Hosen**

---

Yes

Jul 18, 2021, 10:13 PM

---

**Kamal Bharakhda**

---

can u send me this book please

Jul 18, 2021, 10:05 PM

---

**Kamal Bharakhda**

---

means, tub can never be a child of bathroom! but I guess tub is a mehtod

Jul 18, 2021, 10:04 PM



---

due to tub and bathroom analogy

Jul 18, 2021, 10:04 PM

**MD Ismail Hosen**

---

Why so?

Jul 18, 2021, 10:03 PM

**Kamal Bharakhda**

---

interesting as well as confusing....

Jul 18, 2021, 10:03 PM

**MD Ismail Hosen**

---

## Using IS-A and HAS-A

Remember that when one class inherits from another, we say that the subclass *extends* the superclass. When you want to know if one thing should extend another, apply the IS-A test.

Triangle IS-A Shape, yeah, that works.

Cat IS-A Feline, that works too.

Surgeon IS-A Doctor, still good.

Tub extends Bathroom, sounds reasonable.

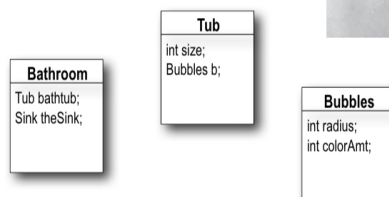
*Until you apply the IS-A test.*

To know if you've designed your types correctly, ask, "Does it make sense to say type X IS-A type Y?" If it doesn't, you know there's something wrong with the design, so if we apply the IS-A test, Tub IS-A Bathroom is definitely false.

What if we reverse it to Bathroom extends Tub? That still doesn't work, Bathroom IS-A Tub doesn't work.

Tub and Bathroom *are* related, but not through inheritance. Tub and Bathroom are joined by a HAS-A relationship. Does it make sense to say "Bathroom HAS-A Tub"? If yes, then it means that Bathroom has a Tub instance variable. In other words, Bathroom has a *reference* to a Tub, but Bathroom does not *extend* Tub and vice-versa.

Does it make sense to say a Tub IS-A Bathroom? Or a Bathroom IS-A Tub? Well it doesn't to me. The relationship between my Tub and my Bathroom is HAS-A. Bathroom HAS-A Tub. That means Bathroom has a Tub instance variable.



Bathroom HAS-A Tub and Tub HAS-A Bubbles.  
But nobody inherits from (extends) anybody else.

Jul 18, 2021, 9:58 PM

Kamal Bharakhda

The following six accessibility levels can be specified using the access modifiers:

- **public:** Access is not restricted. Any caller program can have access to a public member.
- **protected:** Access is limited to the containing class or types derived from the containing class.
- **internal:** Access is limited to the current assembly.
- **protected internal:** Access is limited to the current assembly or types derived from the containing class.
- **private:** Access is limited to the containing type.
- **private protected:** Access is limited to the containing class or types derived from the containing class within the current assembly.

Jul 18, 2021, 8:28 PM



Jul 18, 2021, 7:48 PM

**MD Ismail Hosen**

---

I will do some study on non child

Jul 18, 2021, 7:46 PM

**Kamal Bharakhda**

---

let's say client want to make an application where he can manipulate images

now that app needs two things

1. basic user interface using userform
2. special library that integrates with userform which actually delivers the purpose of app

now, let's say I design 1st activity

then i bundled my work to you in dll and then it's your activity start

Jul 18, 2021, 7:46 PM

**MD Ismail Hosen**

---

Okay

Jul 18, 2021, 7:44 PM

**Kamal Bharakhda**

---

when a library used to build by integrating two different projects

Jul 18, 2021, 7:43 PM

**Kamal Bharakhda**

---

it creates a difference on multiple layer

Jul 18, 2021, 7:42 PM

**MD Ismail Hosen**

---

So still i am confused what's the difference?

Jul 18, 2021, 7:41 PM

**MD Ismail Hosen**

---

---

**MD Ismail Hosen**

Doesn't matter if composition or not

Jul 18, 2021, 7:40 PM

---

**MD Ismail Hosen**

When initializing object you don't add any method or member but use them

Jul 18, 2021, 7:40 PM

---

**Kamal Bharakhda**

it's not a design component but usage component

Jul 18, 2021, 7:37 PM

---

**Kamal Bharakhda**

it only gives facility of USAGE not manipulation

Jul 18, 2021, 7:37 PM

---

**MD Ismail Hosen**

Like school HAS a Students

Jul 18, 2021, 7:36 PM

---

**Kamal Bharakhda**

Class1 obj = New Class1

line will not allow to add or define new members to the obj object

Jul 18, 2021, 7:36 PM

---

**Kamal Bharakhda**

it's not a composition....because

Jul 18, 2021, 7:36 PM

---

**MD Ismail Hosen**

IS a relationship and HAS a relationship

Jul 18, 2021, 7:36 PM

---

**MD Ismail Hosen**

---

**MD Ismail Hosen**

---

But your example is opposite of has a relationship

Jul 18, 2021, 7:35 PM

---

**MD Ismail Hosen**

---

Has a relationship

Jul 18, 2021, 7:35 PM

---

**MD Ismail Hosen**

---

It's called composition

Jul 18, 2021, 7:35 PM

---

**MD Ismail Hosen**

---

Okay. Now it's clear

Jul 18, 2021, 7:34 PM

---

**Kamal Bharakhda**

---

it just increasing the encapsulation

Jul 18, 2021, 7:34 PM

---

**Kamal Bharakhda**

---

yes.... in this case,

now, if owner of the school class, wants to limit users to not to use some methods, it can be done by that...

Jul 18, 2021, 7:34 PM

---

**MD Ismail Hosen**

---

Initializing school class object?

Jul 18, 2021, 7:33 PM

---

**MD Ismail Hosen**

---

Then how Students getting all method or property of school?

Jul 18, 2021, 7:33 PM

---

**Kamal Bharakhda**

---

---

**Kamal Bharakhda**

---

simply it circle around the concept of child and non-child class.

Jul 18, 2021, 7:31 PM

---

**Kamal Bharakhda**

---

here is the different....

inheritance is used by the developers

and non-child class is used by the users

like if you create a library, to do some stuff

you can initiate only non-child class where developer can limit some members by assigning protected specifier

Jul 18, 2021, 7:30 PM

---

**MD Ismail Hosen**

---

Then it's composition

Jul 18, 2021, 7:29 PM

---

**MD Ismail Hosen**

---

Okay

Jul 18, 2021, 7:29 PM

---

**Kamal Bharakhda**

---

because, it is not inherited

Jul 18, 2021, 7:29 PM

---

**MD Ismail Hosen**

---

It's inheriting all of school

Jul 18, 2021, 7:29 PM

---

**MD Ismail Hosen**

---

Why non child?

Jul 18, 2021, 7:28 PM

---

**Kamal Bharakhda**

---

now, anywhere



myStudent object will be non-child class of School

Jul 18, 2021, 7:28 PM

---

**MD Ismail Hosen**

But is constructor overriable?

Jul 18, 2021, 7:28 PM

---

**MD Ismail Hosen**

And you are overriding it.

Jul 18, 2021, 7:28 PM

---

**MD Ismail Hosen**

Then School is constructor of School class

Jul 18, 2021, 7:27 PM

---

**MD Ismail Hosen**

Got it

Jul 18, 2021, 7:27 PM

---

**Kamal Bharakhda**

yes

Jul 18, 2021, 7:26 PM

---

**Kamal Bharakhda**

School is base class...  
and Students is child class... and we are inheriting all the members of school in student class

Jul 18, 2021, 7:26 PM

---

**MD Ismail Hosen**

Student is a child of School or something right?

Jul 18, 2021, 7:26 PM

---

**MD Ismail Hosen**

School is a constructor or ILB

Jul 18, 2021, 7:25 PM

Autonomous class or something like that

Jul 18, 2021, 7:25 PM

---

**Kamal Bharakhda**

what above code is doing can u tell me ?

Jul 18, 2021, 7:25 PM

---

**Kamal Bharakhda**

```
Public Class Students : School {  
}
```

Jul 18, 2021, 7:25 PM

---

**MD Ismail Hosen**

In java we can have class within class

Jul 18, 2021, 7:25 PM

---

**MD Ismail Hosen**

Is it similar to composition or internal class?

Jul 18, 2021, 7:24 PM

---

**MD Ismail Hosen**

I am not clear about non child class

Jul 18, 2021, 7:24 PM

---

**MD Ismail Hosen**

Okay

Jul 18, 2021, 7:23 PM

---

**Kamal Bharakhda**

so, protected members will only be able to visible through same class as well as child class..  
to understand protected internal we have to understand first internal

internal members cannot be visible on other projects but it can be visible within same namespace or  
project through all the class

so in that case, protected internal would be a combination of protected and internal members,

Protected internal members could be visible from same project as well different object but only using  
child class.. not allowed in non-child class

Jul 18, 2021, 7:22 PM

What could be a use case?

Jul 18, 2021, 7:19 PM

---

**MD Ismail Hosen**

Interesting

Jul 18, 2021, 7:19 PM

---

**Kamal Bharakhda**

child class is inheritated from the base class.

when class is assigned to a object using constructor method, we call that object as non-child class

Jul 18, 2021, 7:18 PM

---

**Kamal Bharakhda**

do you know the diff between, child class and non-child class?

Jul 18, 2021, 7:17 PM

---

**MD Ismail Hosen**

What is the difference between protected and protected internal?

Jul 18, 2021, 7:16 PM

---

**MD Ismail Hosen**

Okay

Jul 18, 2021, 7:14 PM

---

**Kamal Bharakhda**

nope not required...

Jul 18, 2021, 7:14 PM

---

**MD Ismail Hosen**

Want more explanation?

Jul 18, 2021, 7:14 PM

---

**MD Ismail Hosen**

In const you need to specify at design time

Jul 18, 2021, 7:12 PM

You don't need to specify it. You can assign at run time

Jul 18, 2021, 7:12 PM

---

**Kamal Bharakhda**

but then how it is different then const ?

Jul 18, 2021, 7:11 PM

---

**Kamal Bharakhda**

ok....

Jul 18, 2021, 7:11 PM

---

**MD Ismail Hosen**

Once set you can't change it

Jul 18, 2021, 7:11 PM

---

**Kamal Bharakhda**

ok thank you....

thanks a lot brother..... our group has cleared a lot of concepts in OOP, that's why I'm now being able to take over any framework and language.

I found C# a really cool language

Jul 18, 2021, 7:11 PM

---

**MD Ismail Hosen**

It's like const but not constant

Jul 18, 2021, 7:11 PM

---

**MD Ismail Hosen**

Final is very interesting

Jul 18, 2021, 7:10 PM

---

**MD Ismail Hosen**

I guess this is all

Jul 18, 2021, 7:08 PM

---

**MD Ismail Hosen**

Protected



---

**MD Ismail Hosen**

I haven't count. Private,public,final

Jul 18, 2021, 7:08 PM

---

**MD Ismail Hosen**

No problem

Jul 18, 2021, 7:05 PM

---

**Kamal Bharakhda**

How many Access Specifiers in Java

In C# there are 5

private  
internal  
protected  
protected internal  
public

Jul 18, 2021, 7:05 PM

---

**MD Ismail Hosen**

Yes

Jul 18, 2021, 7:04 PM

---

**Kamal Bharakhda**

is it right time to discuss with you?

Jul 18, 2021, 7:03 PM

---

**MD Ismail Hosen**

It just sad but nothing we can do about it

Jul 18, 2021, 7:03 PM

---

**MD Ismail Hosen**

Yeah

Jul 18, 2021, 7:03 PM

---

**Kamal Bharakhda**

so I'm sure you will handle the crisis well...

Jul 18, 2021, 7:03 PM