

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

For example, Triangle **IS-A** Shape, that works; Cat **IS-A** Feline, that works too. Tub extends Bathroom, sounds reasonable. **Until you apply the IS-A test**.

To know if we have designed our types correctly, ask, “Does it make sense to say type X IS-A type Y?” If it doesn’t, we 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.

## Examples

**IS-A:** Cat is a Feline

```
class Feline {
...
}

class Cat extends Feline {
...
}
```

**HAS-A:** Bathroom has a Tub

```
class Bathroom {
Tub newTub = new Tub();
...
}
```