Misused of Inheritance

Ask yourself: "Am I using inheritance to simply share attributes or behavior without further adding anything special in my subclasses?"

Example of misused Inheritance

```java
public class Pepperoni extends Pizza {

    public Pepperoni(String size,
    String crust) {
        super(size, crust);
        super.addTopping("pepperoni");
    }
}
```

Here Super(base class) already all attributes , in above sub class you calling base call & calling function is misused of inheritances. Here subclass is not different from base class

This is base class

```java
public class Pizza {
    private List toppings;
    private String size;
    private String crustStyle;

    public Pizza(String size, String crust) {
        this.toppings = new ArrayList();
        this.size = size;
        this.crustStyle = crust;
    }

    public void addTopping(String topping) {
        this.toppings.add(topping);
    }

    public void bulkAddTopping(ArrayList toppingList) {
        this.toppings.addAll(toppingList);
    }

    public void cook() throws InterruptedException {
        wait(10 * 6000);
    }
}
```

Given the following superclass-subclass pairings, which one is a proper use of inheritance?

☐ Superclass: Sandwich. Subclass: Ham; Tuna Salad; Chicken Salad

☑ Superclass: Animal. Subclass: Dog, Cat, Whale

> ⊘ **Correct**
> Nice job! There is probably some behaviour that is common to all of these, and some behaviour that is very different between them.

☑ Superclass: Employee Subclass: Manager, Salesperson, Cashier

> ⊘ **Correct**
> Great! Employee is a generalized type for manager, salesperson, and cashier, but each of these types of employee perform specific functions.

☑ Superclass: Dog   Subclass: Husky, Golden Retriever, Poodle

> ⊗ **This should not be selected**
> Not this one. Although the dogs are different, their behaviours are the same, so the superclass is enough!

Subclass behavior should be different.