CYDEO

Abstraction

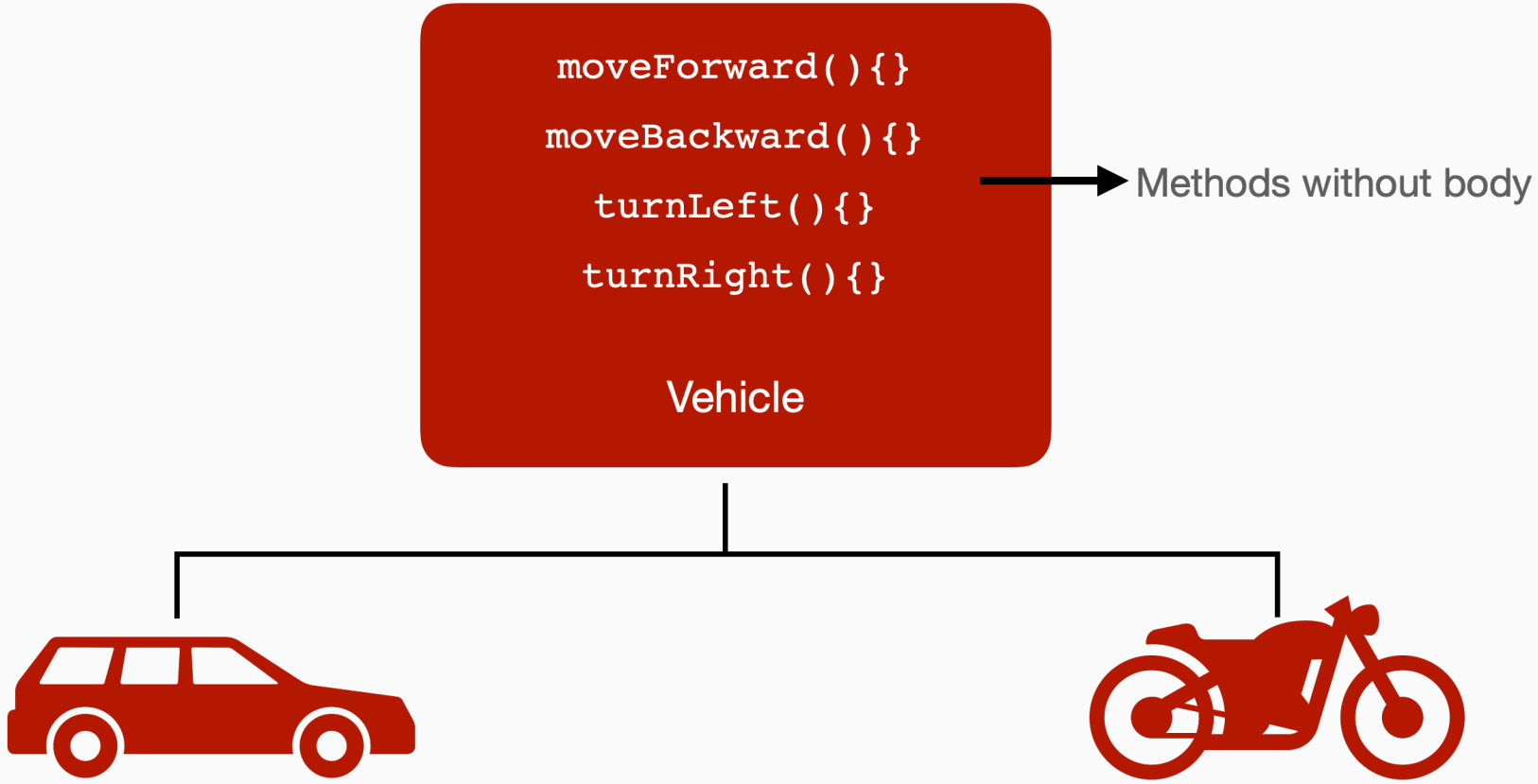
- Abstract Method
- Abstract Class

OOP Principles

- There are 4 Object Oriented Programming (OOP) principles:
    - Encapsulation
    - Inheritance
    - Abstraction
    - Polymorphism

What Is Abstraction?

- Focus only on relevant properties of the problem
- 'Ignore' details.



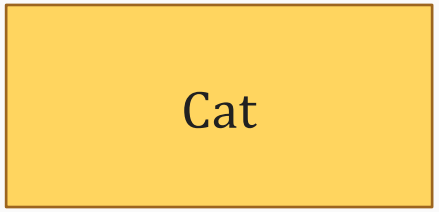
Abstraction

- Process of hiding implementation details from the user
- Only the functionality will be provided to the user
- Focusing on the essential qualities of something rather than one specific example. (Ignoring the irrelevant & unimportant)
- User will have the information on what the object does instead of how it does

Abstraction

eat():

eats dog food



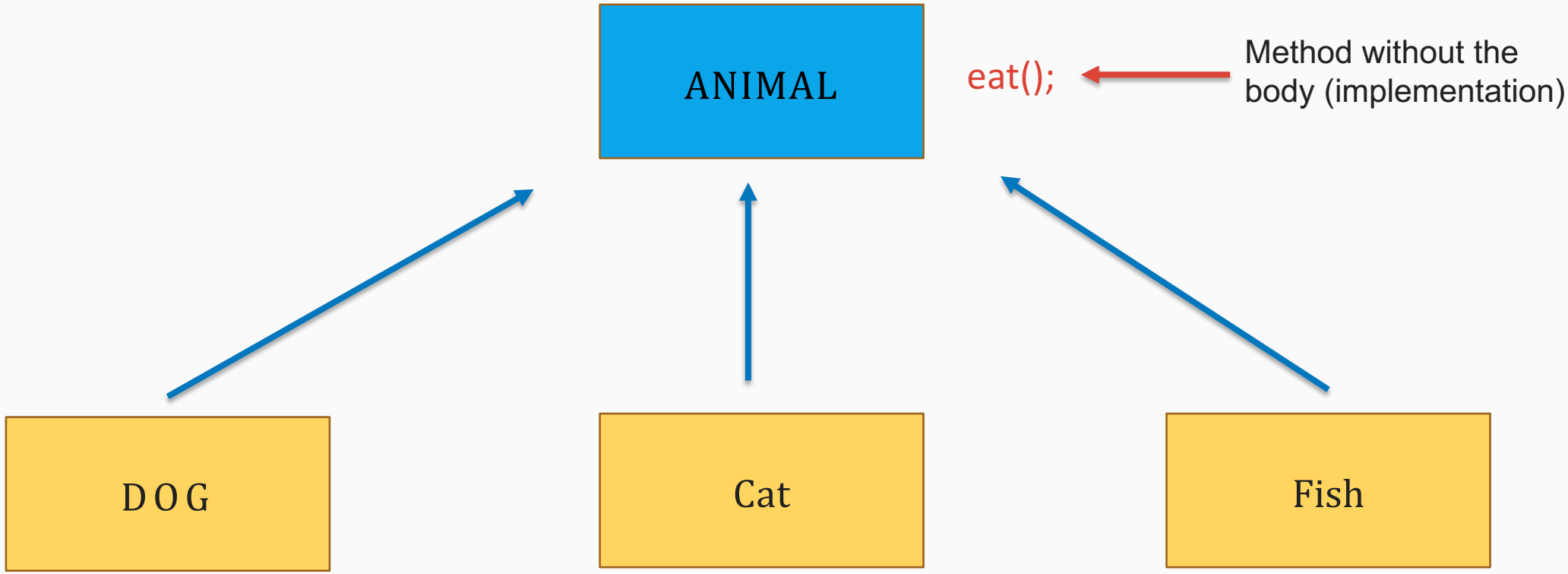
eat():

eats cat food

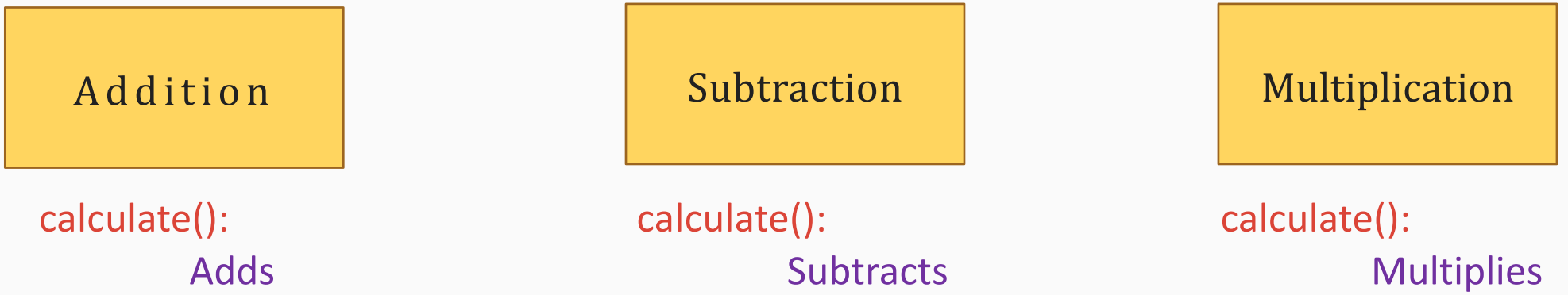
eat():

eats fish food

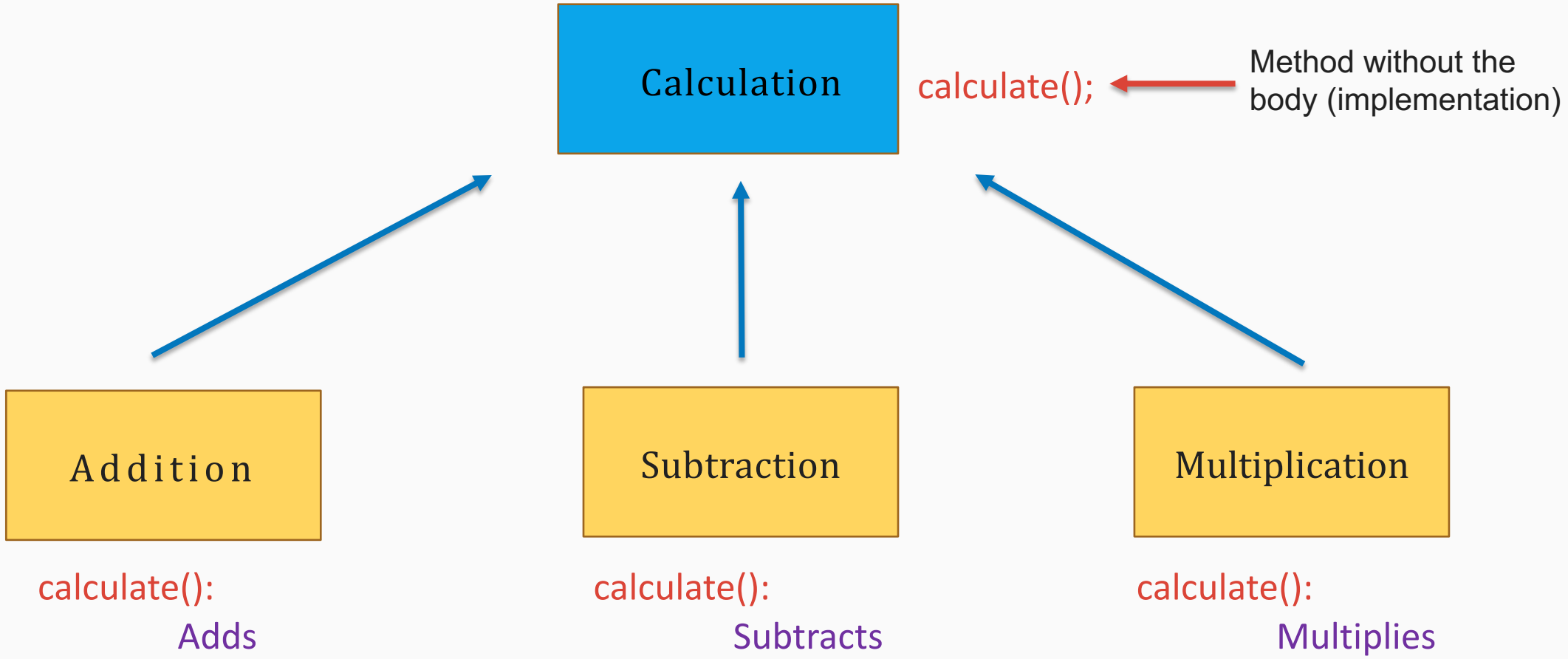
Abstraction



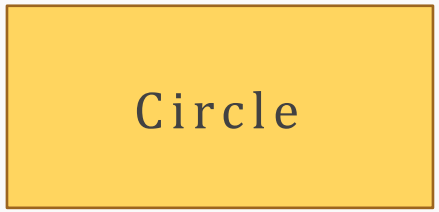
Abstraction



Abstraction

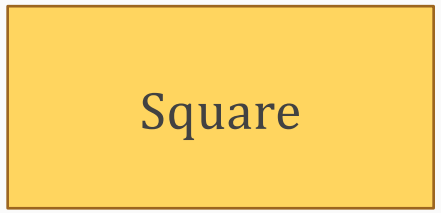


Abstraction



area():

radius * radius * pi



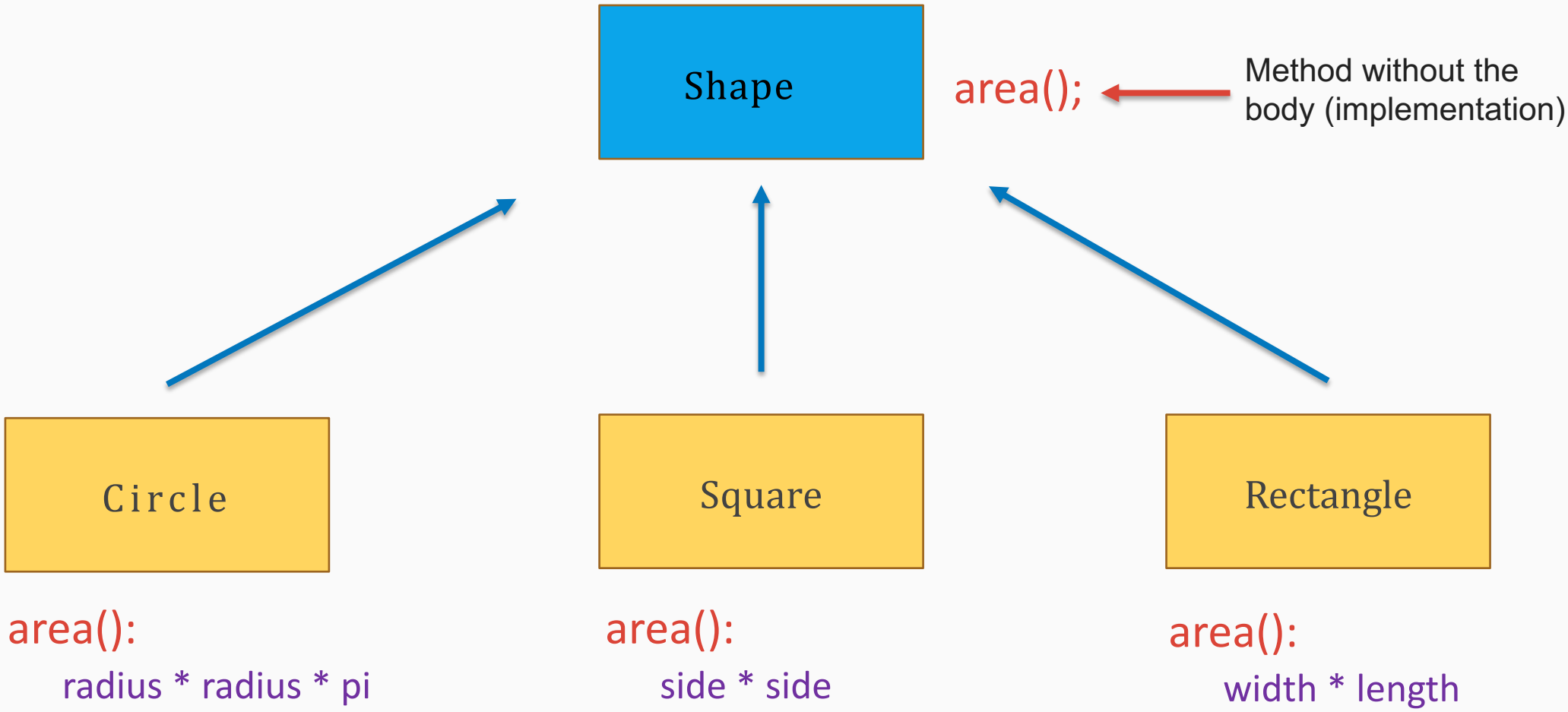
area():

side * side

area():

width * length

Abstraction



Abstract Method

- A method that does not have body, only signature
- A method that's meant to be overridden
- The abstract keyword is used to create abstract method

```
public abstract void eat();
```

Method without the body (implementation)

```
public abstract double area();
```

Method without the body (implementation)

CYDEO

Abstract Method Rules

- An abstract method can not be static
- An abstract method can not be final
- An abstract method can not have private access modifier
- An abstract method does not have body
- An abstract method can only be created in an abstract class or in an interface

Abstract Class

- A class that's meant to be a parent (super) class
- Goal is to provide reusable variables and methods to sub classes
- The abstract keyword is used to create the abstract class
- An abstract class can not be instantiated

```
public abstract class Animal{


}
```

CYDEO

Abstract class & Abstract Method

```java
public abstract class Animal{

    public String breed;
    public char gender;

    public Animal(String breed, char gender){
        this.breed = breed;
        this.gender = gender;
    }

    public abstract void eat();

}
```

Abstract class

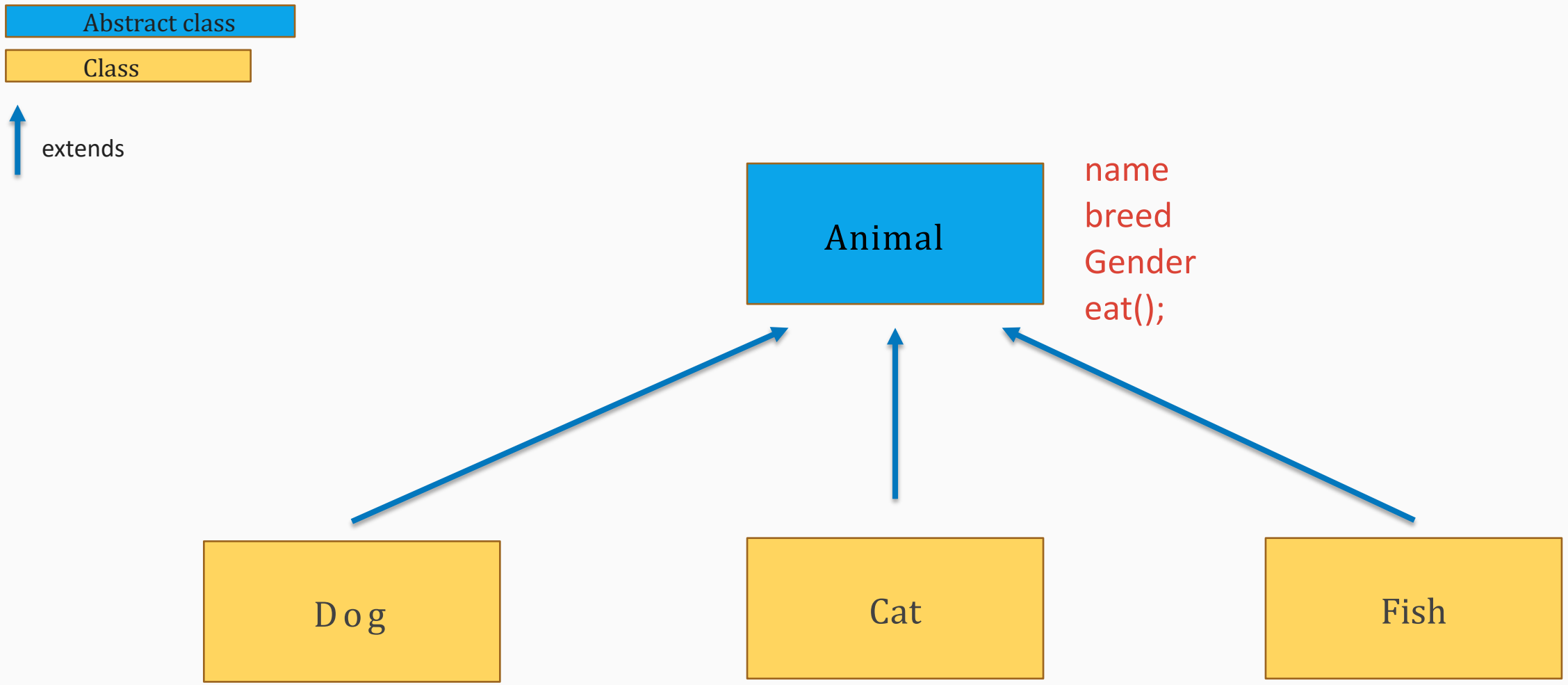
Super (parent) class is responsible to provide the variables and methods that are needed to the all the sub classes without worrying about the small details

Abstract class

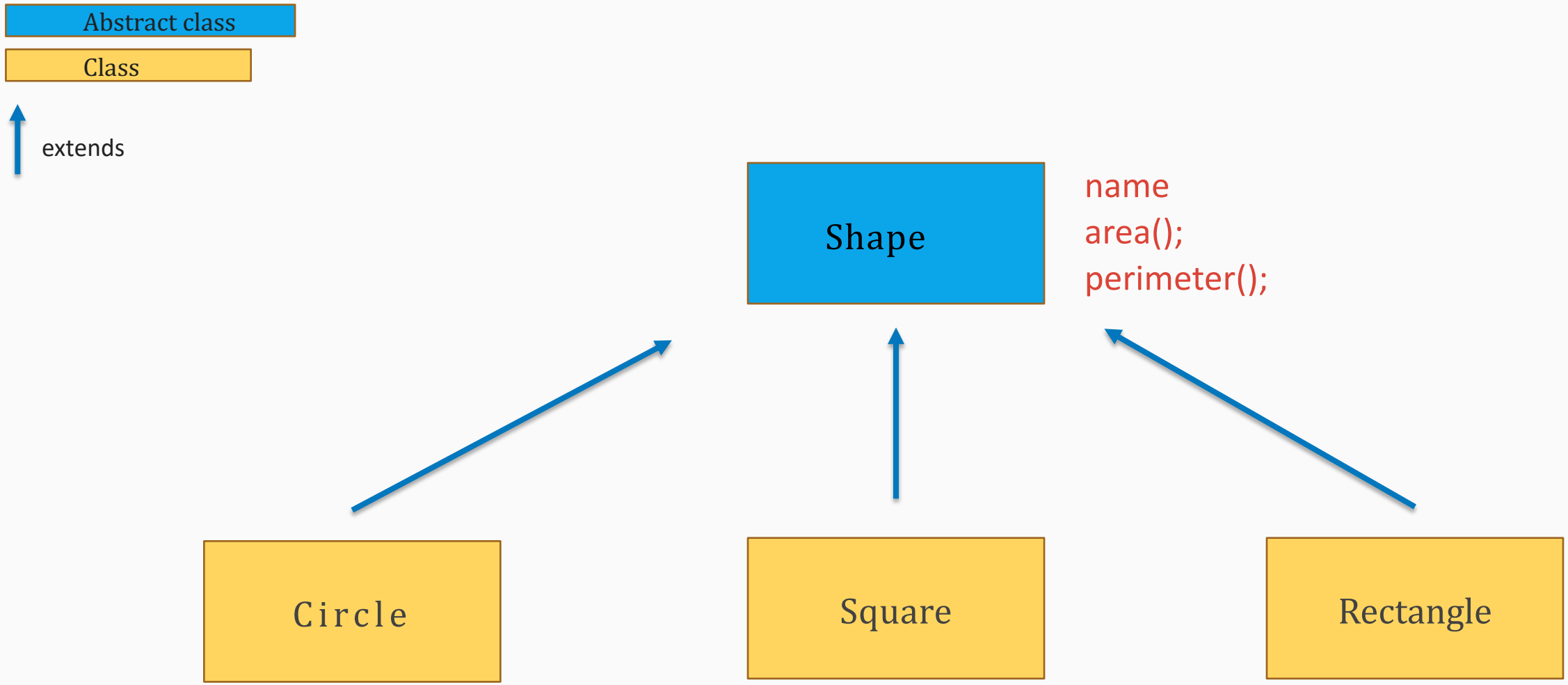
⚠ Sub (child) classes are responsible for providing the implementations that are needed

CYDEO

Abstraction



Abstraction



Creating Object

- Abstract class is meant to be inherited only, not meant to be instantiated
- Abstract class is not a concrete class and object has to be concrete
- A sub class of abstract class is called concrete class, and it can be instantiated
- A Concrete class must implement all the inherited abstract methods

```
public abstract class Animal{

    public abstract void eat();

}

public class Dog extends Animal{

    @Override
    public void eat(){
        System.out.println("Dog eats dog foods");
    }


}
```

Abstract Class vs Concrete class

Regular class	Abstract class
can have constructors, instances and statics	can have constructors, instances and statics
Regular class can be instantiated	Abstract class can not be instantiated
Regular class can not have abstract method	Abstract class can have abstract method
Regular class can be declared as final	Abstract class can not be declared as final