Class & Object

Class & Object

Class: A blueprint for creating objects. No memory allocated for a class

Object: An instance of a class. Each Object has its own memory

Class

```
public class Book{
    String name;
    String author;
}
```

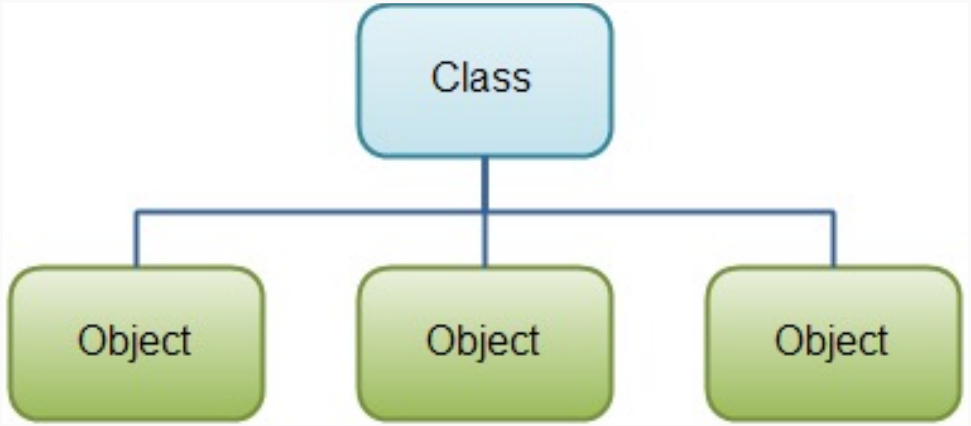
Object

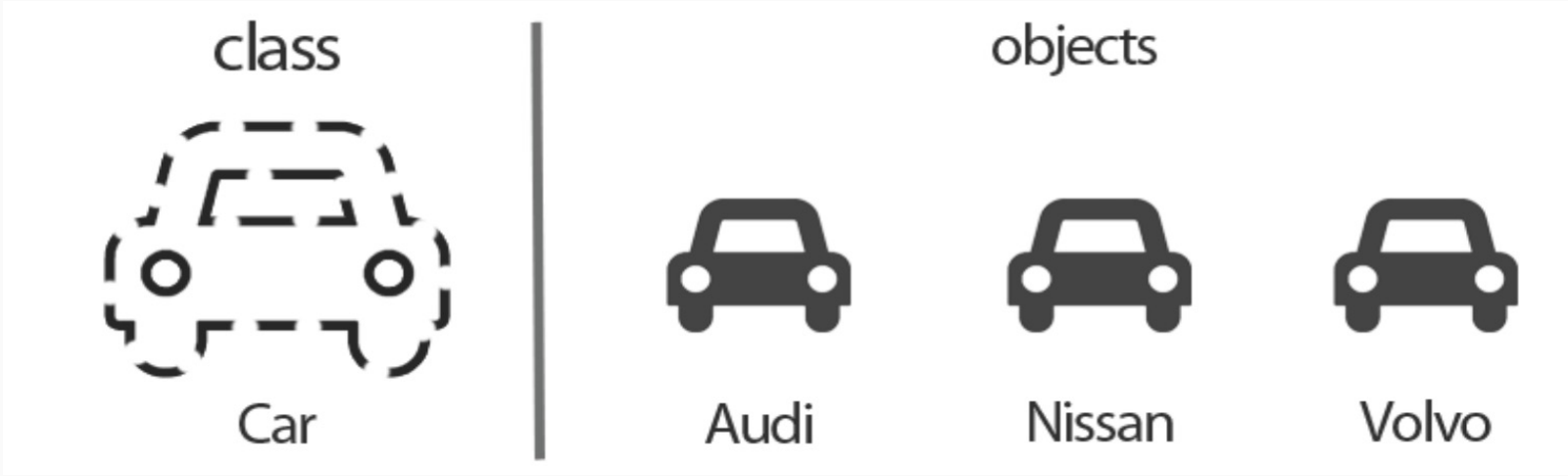
```
Book objBook = new Book();
```

CYDEO

Class & Object

Class is where all objects came from. Object can not exist without the class

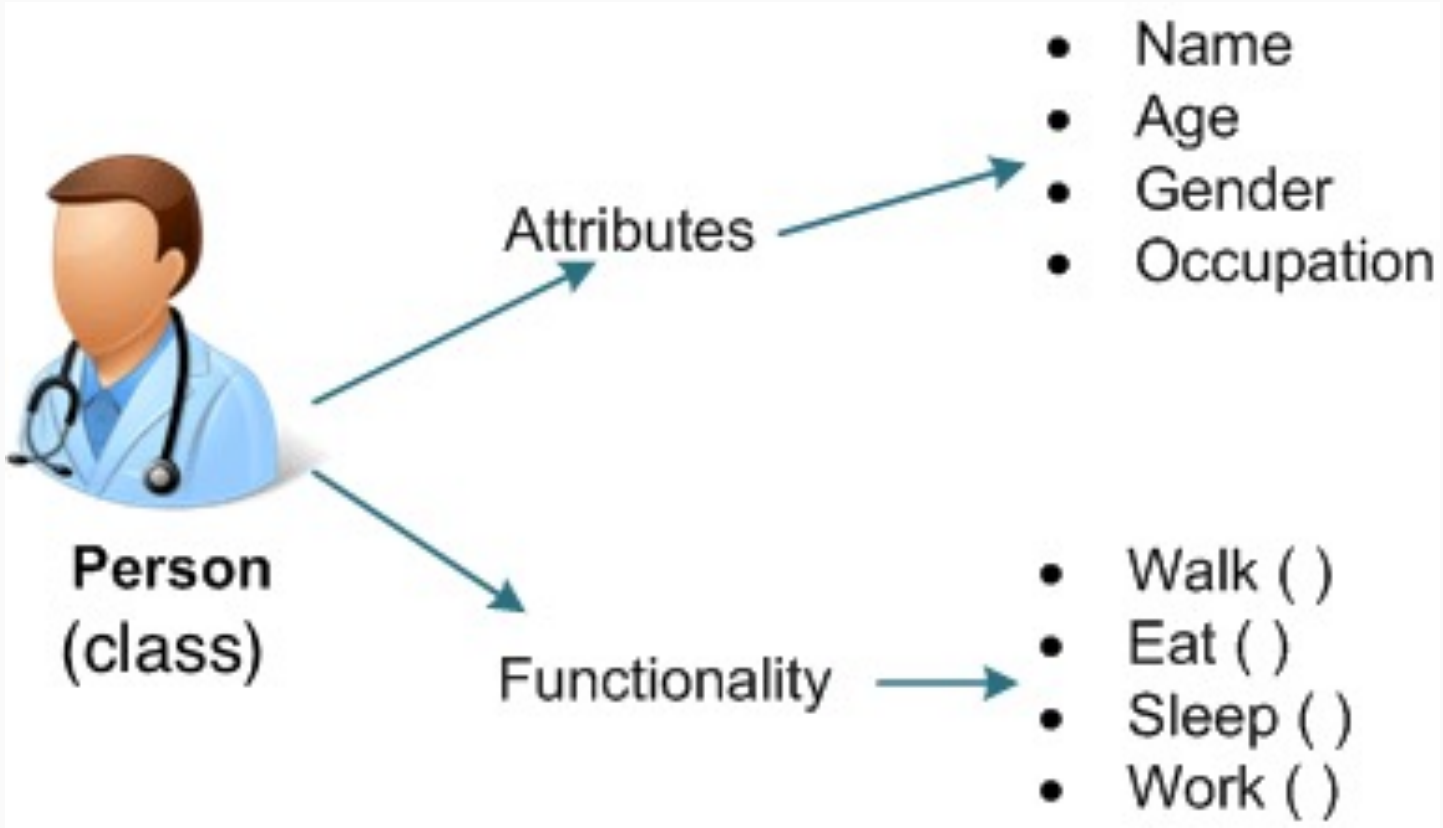


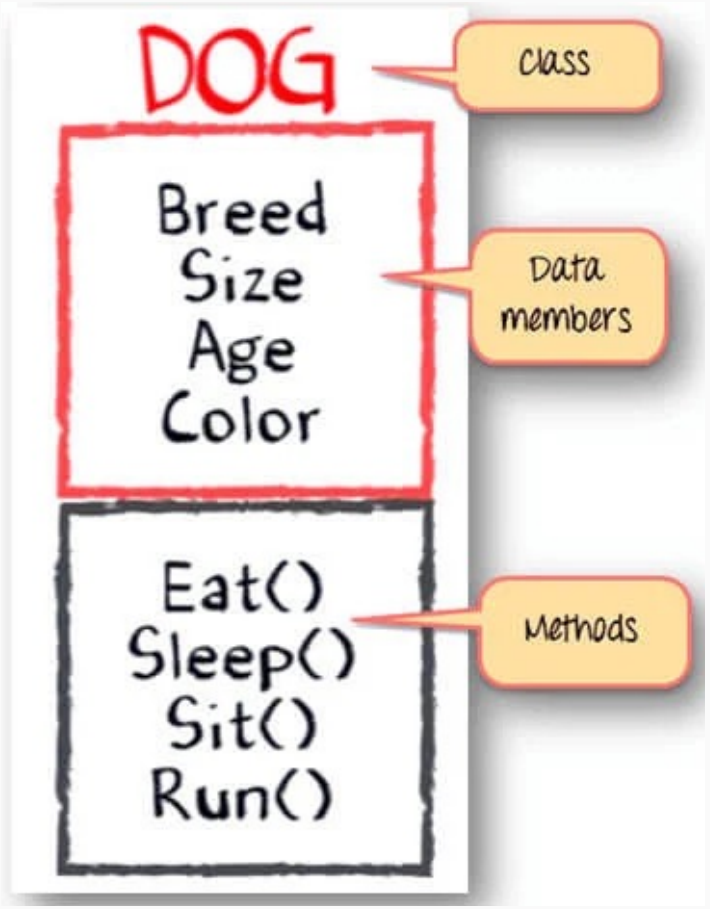


CYDEO

Class & Object

Class determines how an object will behave and what the object will contain

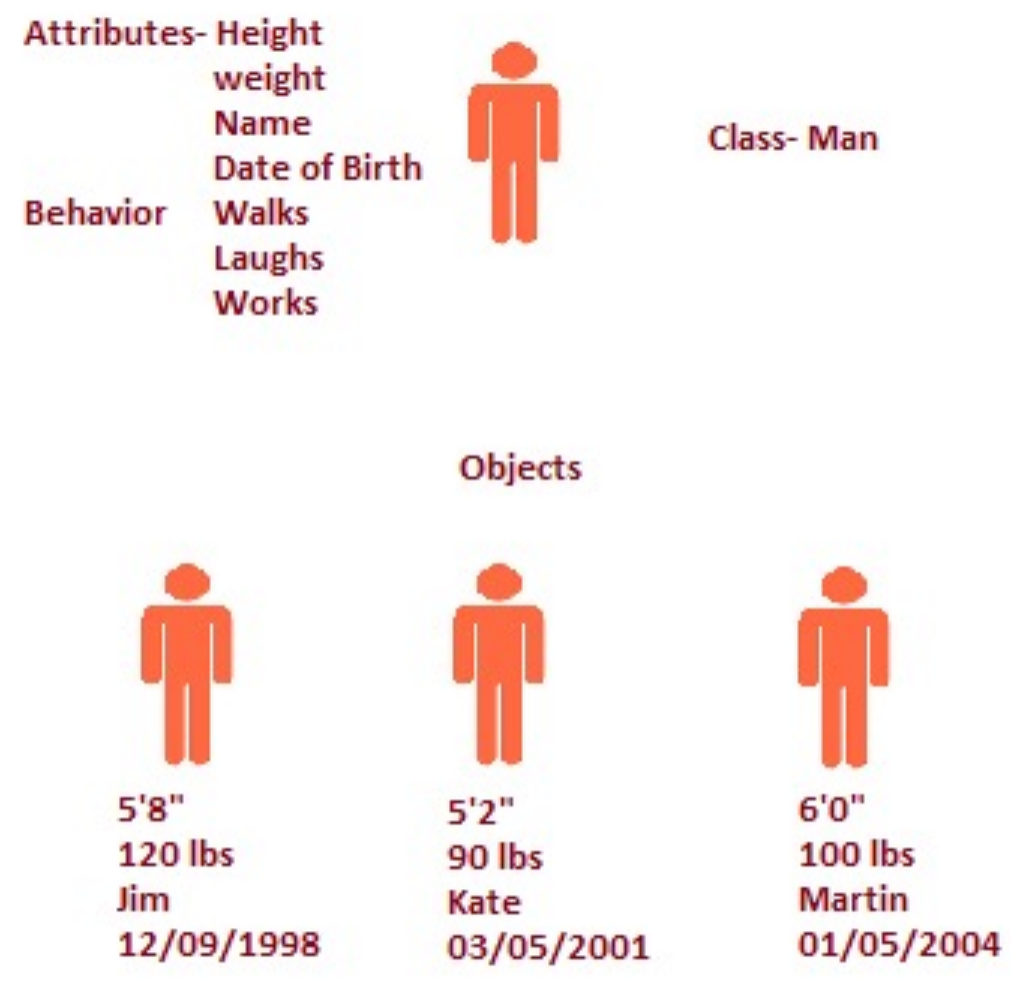


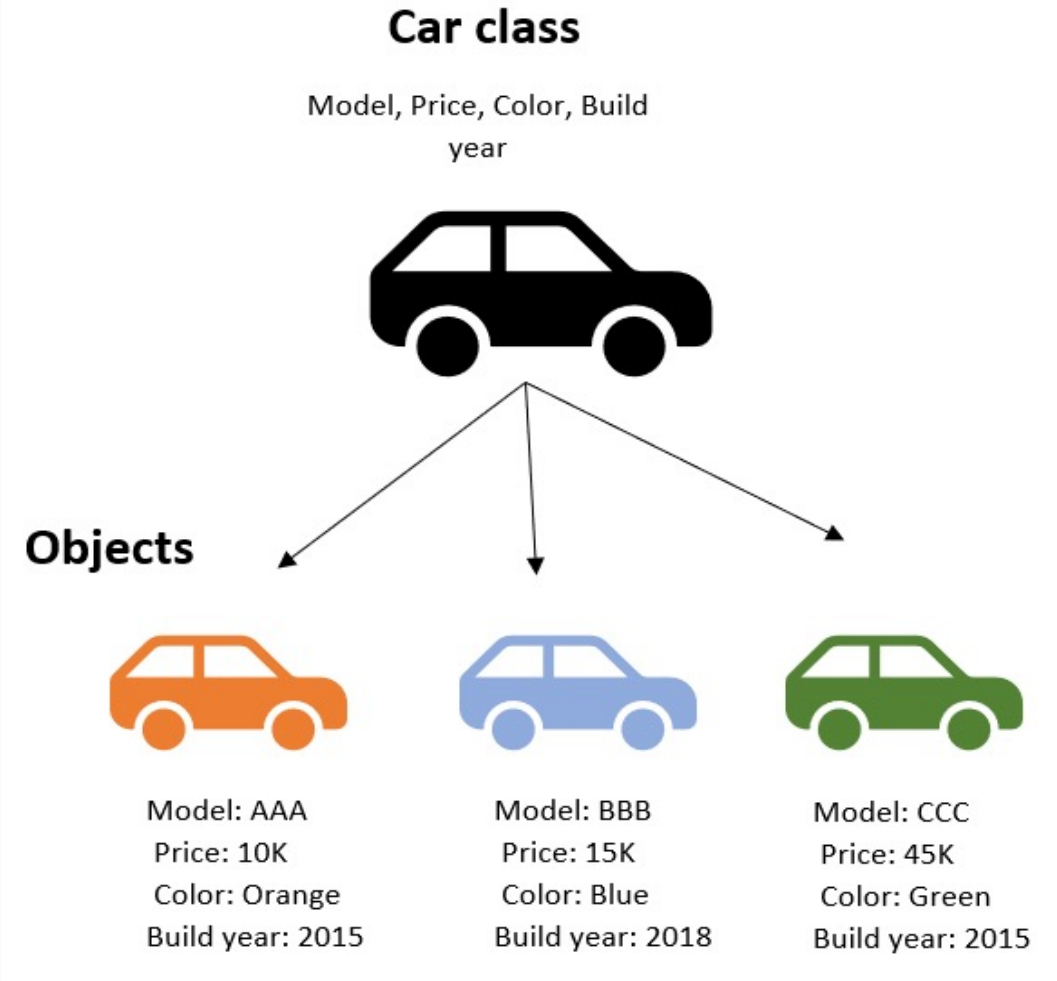


CYDEO

Class & Object

Multiple objects can be created from a class, each object has a different memory



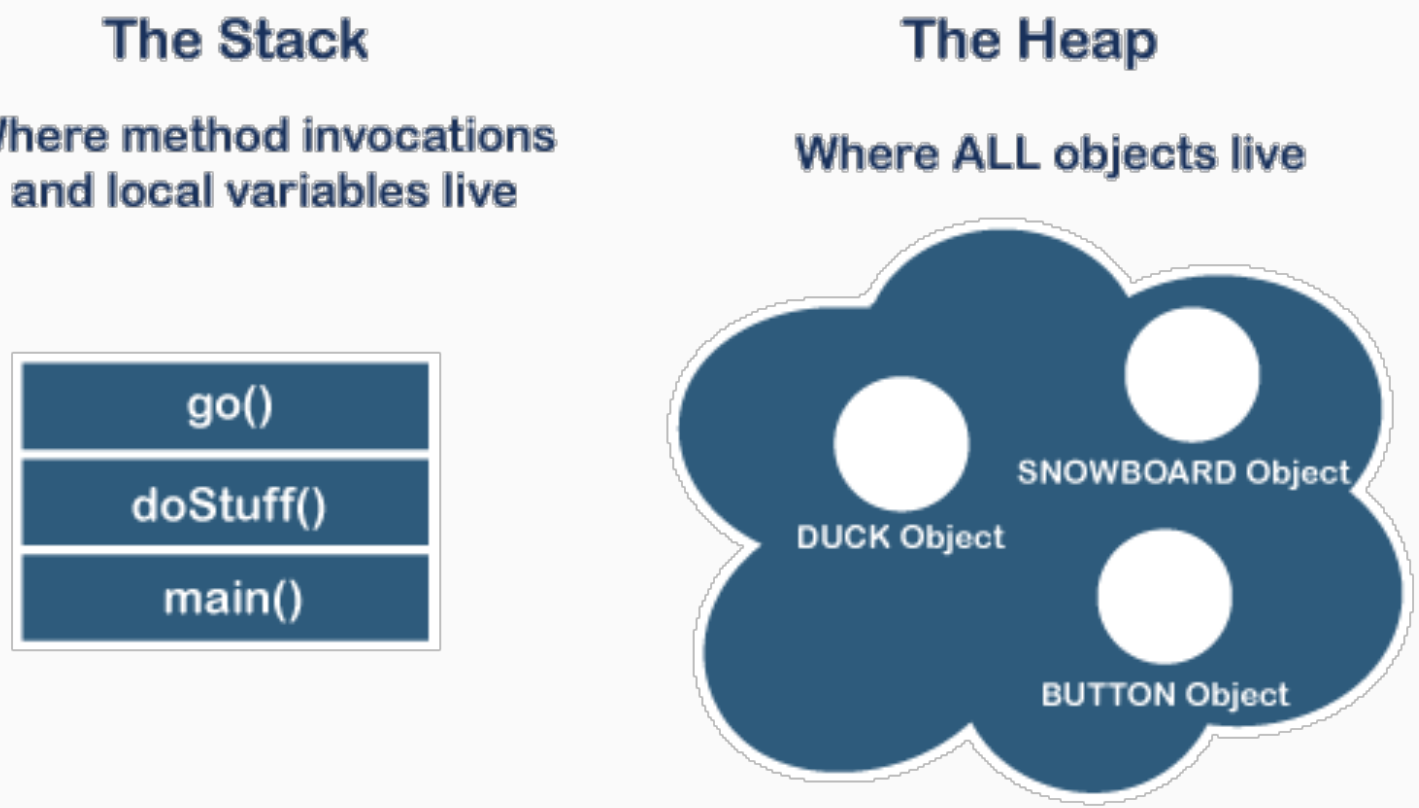


CYDEO

Memory Management

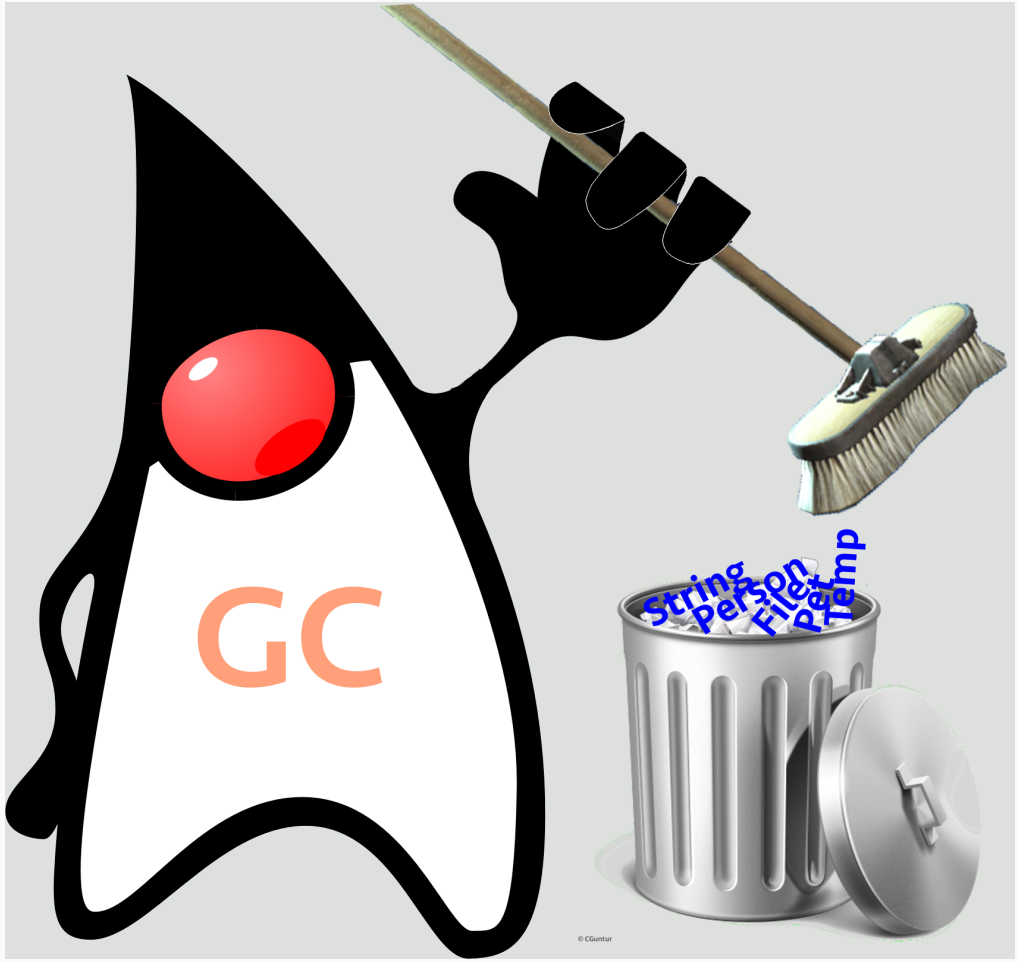
Memory Allocation

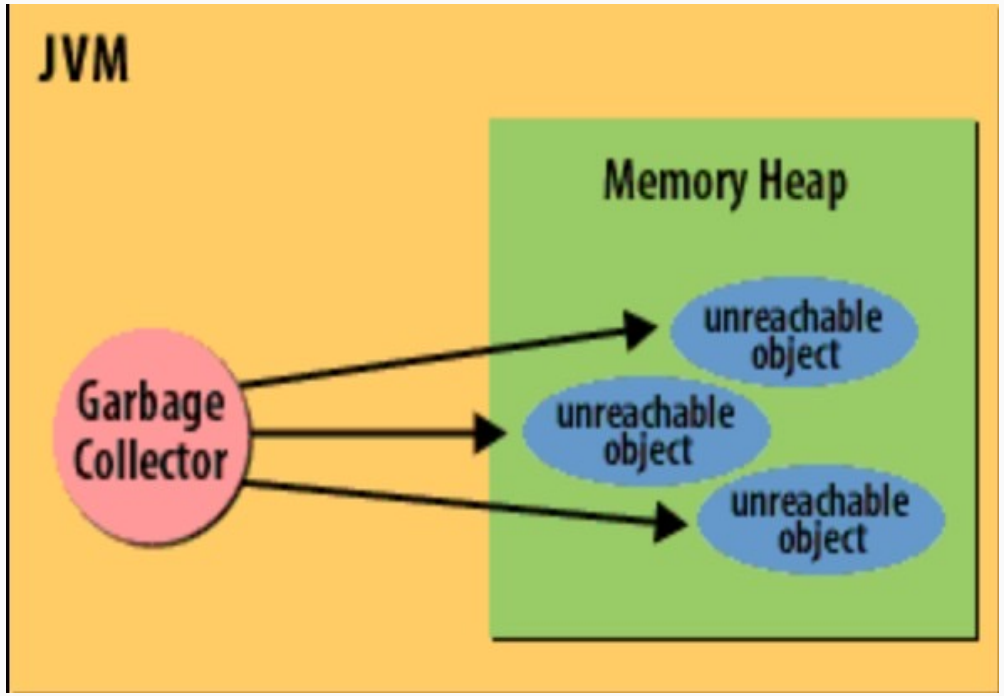
Process of allocation of objects, JVM divides memory into stack and heap



Memory Deallocation

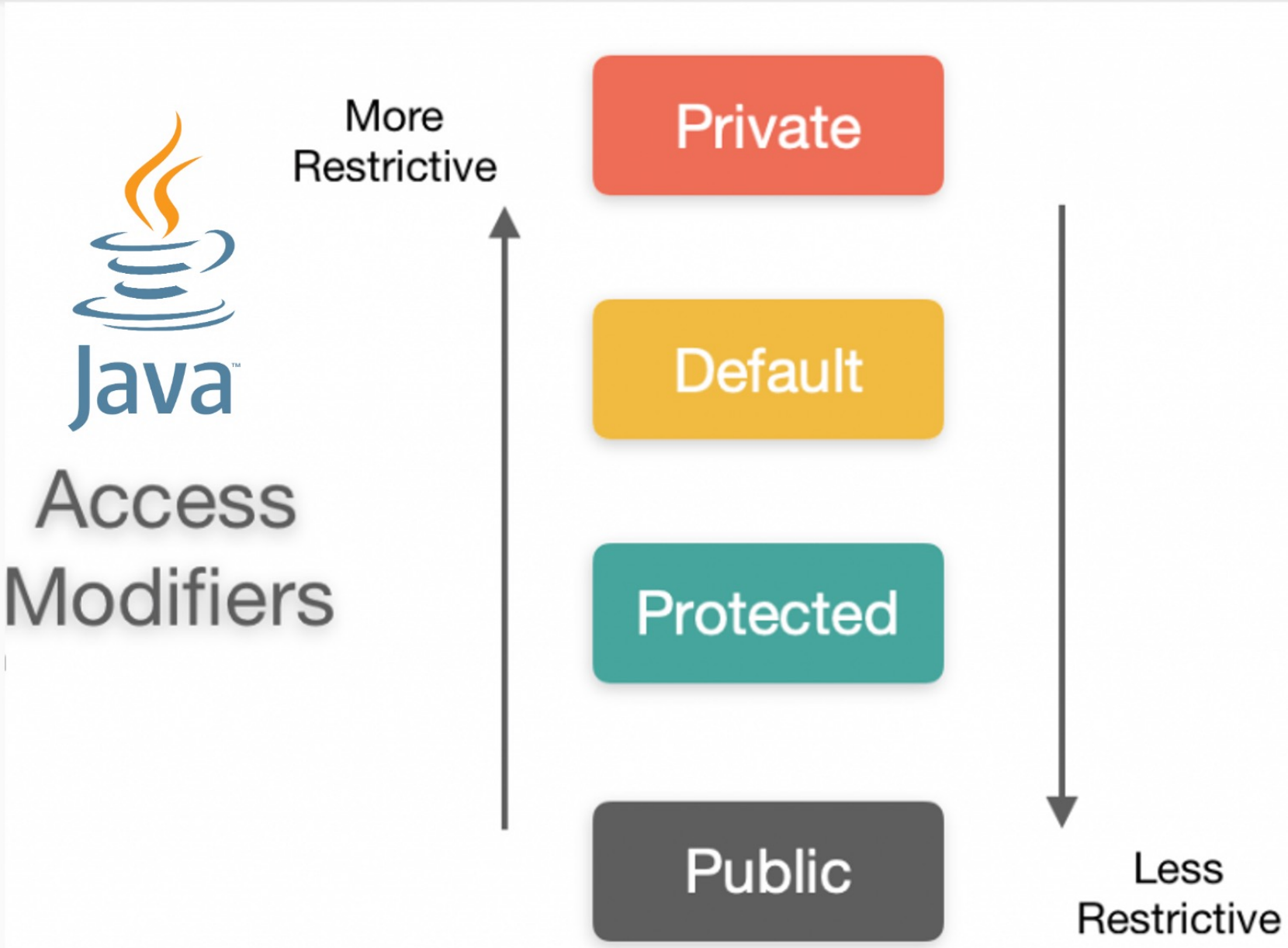
Process of allocation of objects, JVM periodically runs a process known as the garbage collector, which removes unreferenced objects from heap memory





Access Modifiers

What Are the Access Modifiers in Java?



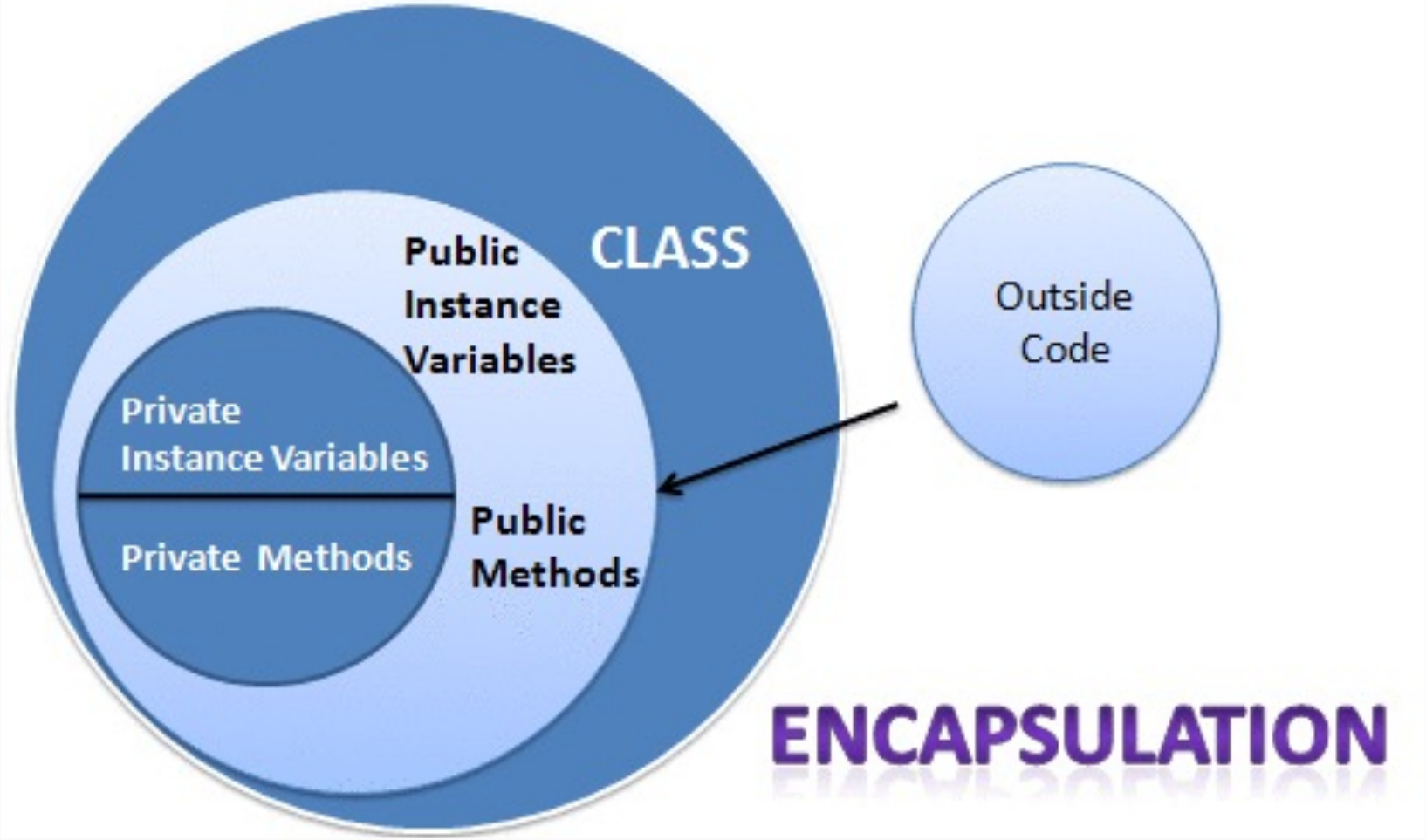
Access Modifiers

Access Modifiers	class	package	Subclass	World
public	Yes	Yes	Yes	Yes
protected	Yes	Yes	Yes	No
default (no modifier)	Yes	Yes	No	No
private	Yes	No	No	No

OOP Encapsulation

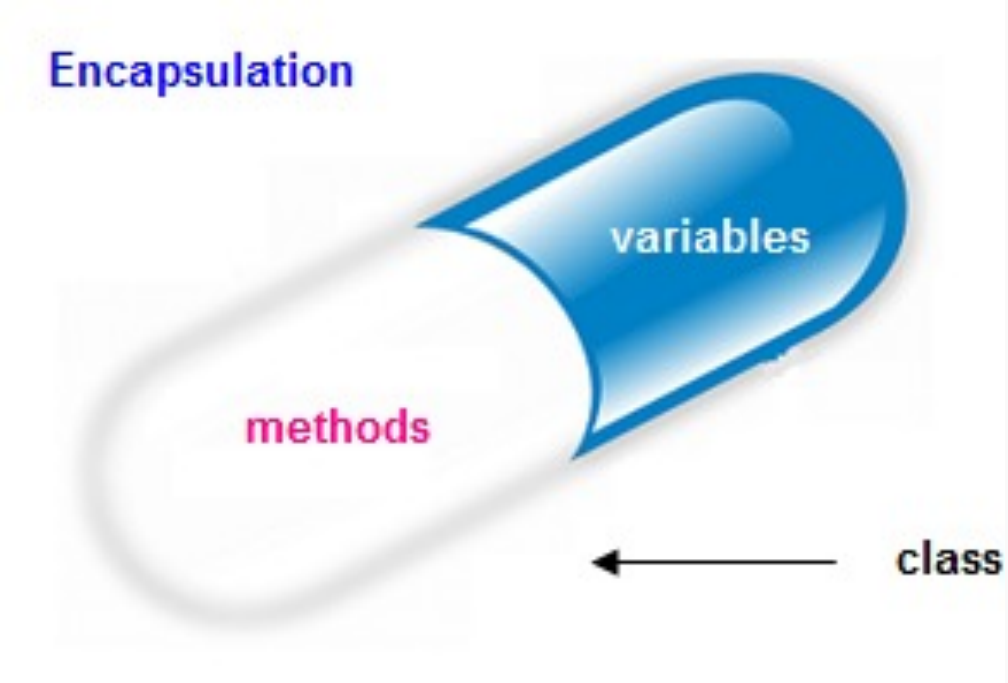
What Is Encapsulation in Java?

- An object hides its internal data from code that's outside the class



Encapsulation (Data Hiding)

- Only the current class' methods can directly access and make changes to the instance variables
- Hide an instance variable by giving private access modifier, and making the methods that access those fields public
- These public methods are called getters & setters (accessor and mutator)



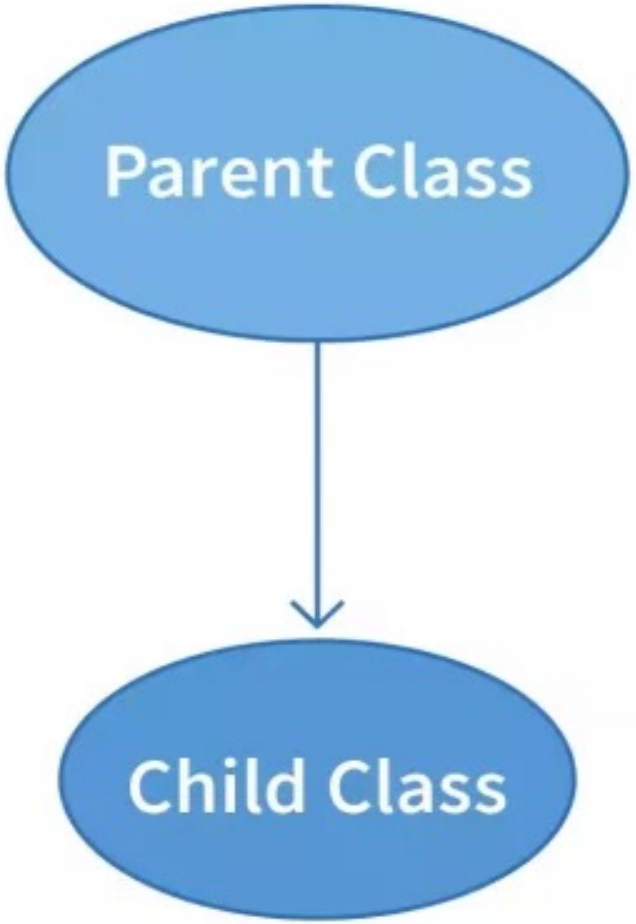
CYDEO

Getters & Setters

- Both are public instance methods, used to protect our data and make our code more secure
- Getter is used for reading the private data (instance variable) only
- Setter is used for writing (modifying) the private data (instance variable) only

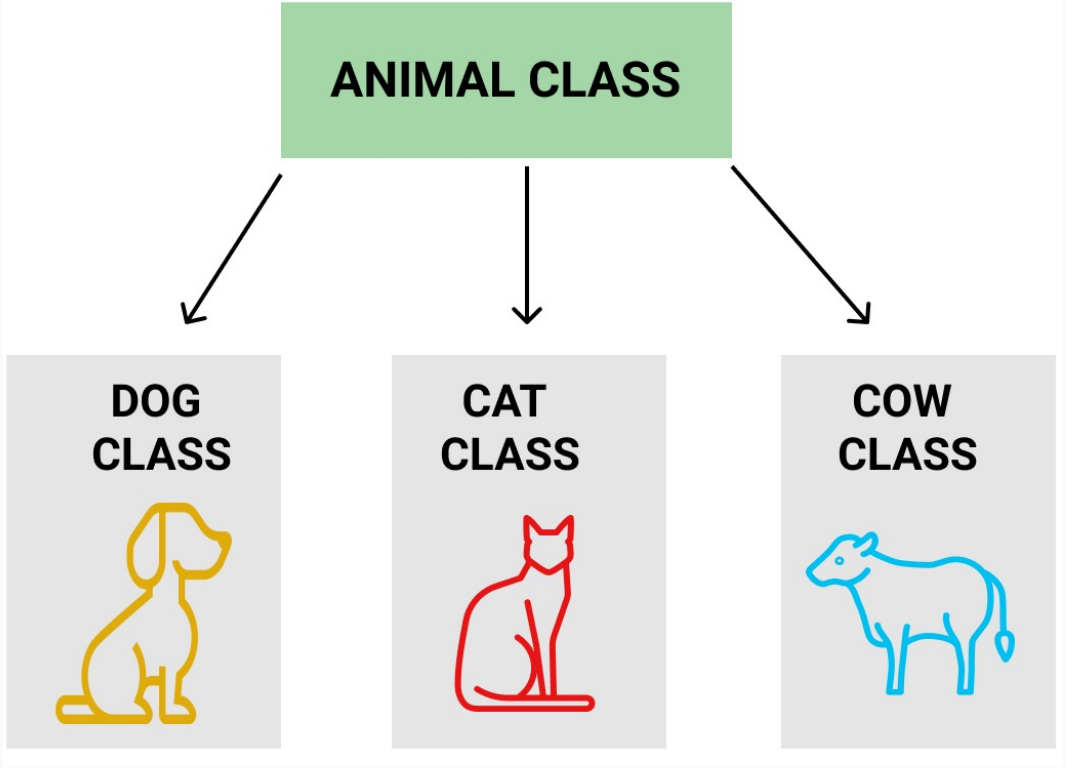
OOP Inheritance

What Is Inheritance?



Inheritance (Is A relation)

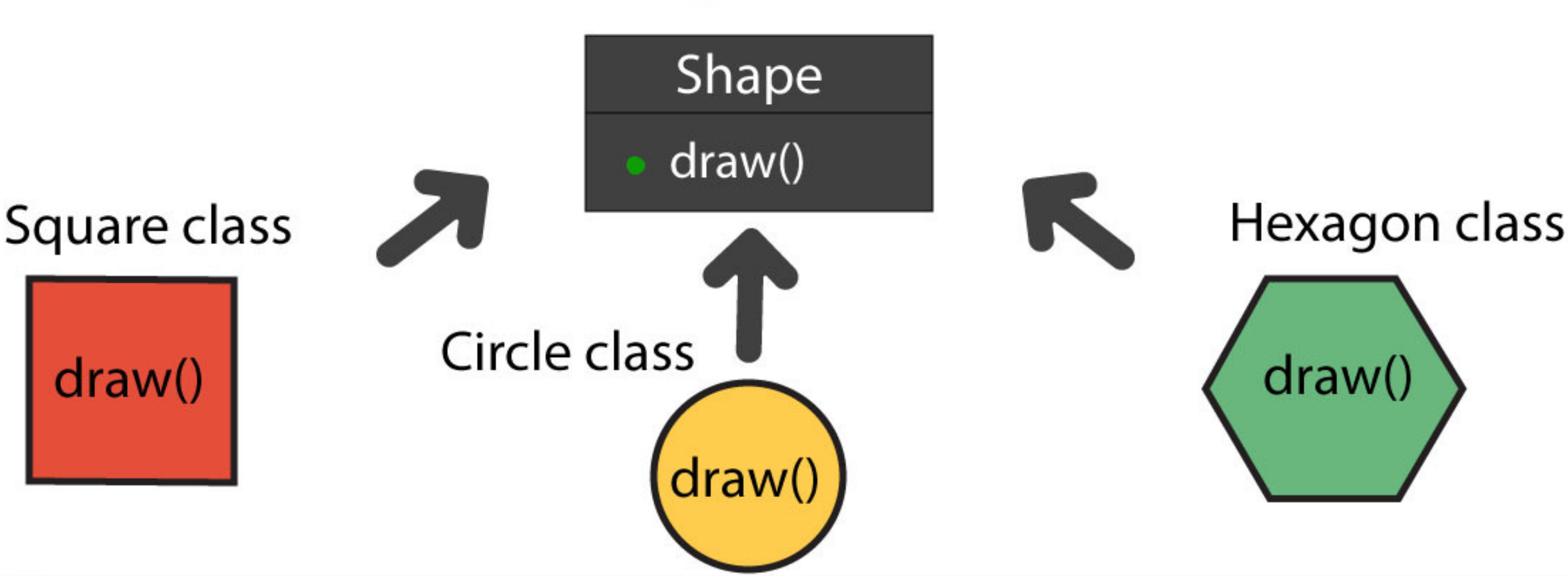
- Used for creating an "is a" relationship among the classes
- When an "is-a" relationship exists between objects, it means that the specialized object has all the characteristics of the general object.
- It allows one class to inherit the features (variables & methods) of another class



CYDEO

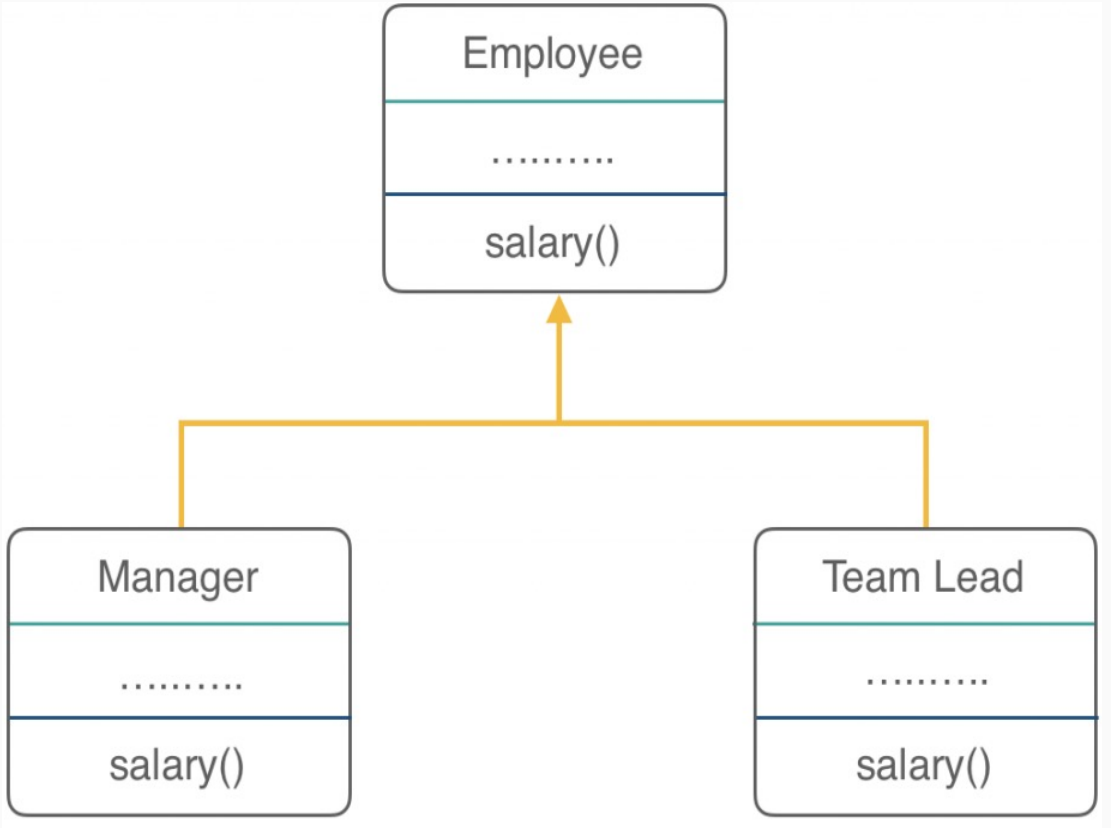
Overriding

What Is Method Overriding?



Method Overriding

- Giving different implementations to the method
- One method having multiple different implementations
- Overriding a method must take place in subclass
- Less memory usage and Improves the reusability of our code



CYDEO

Final Keyword

What Is Final Keyword in Java?

- Used to restrict the user

Final Variable

Can't Re- assign

Final Method

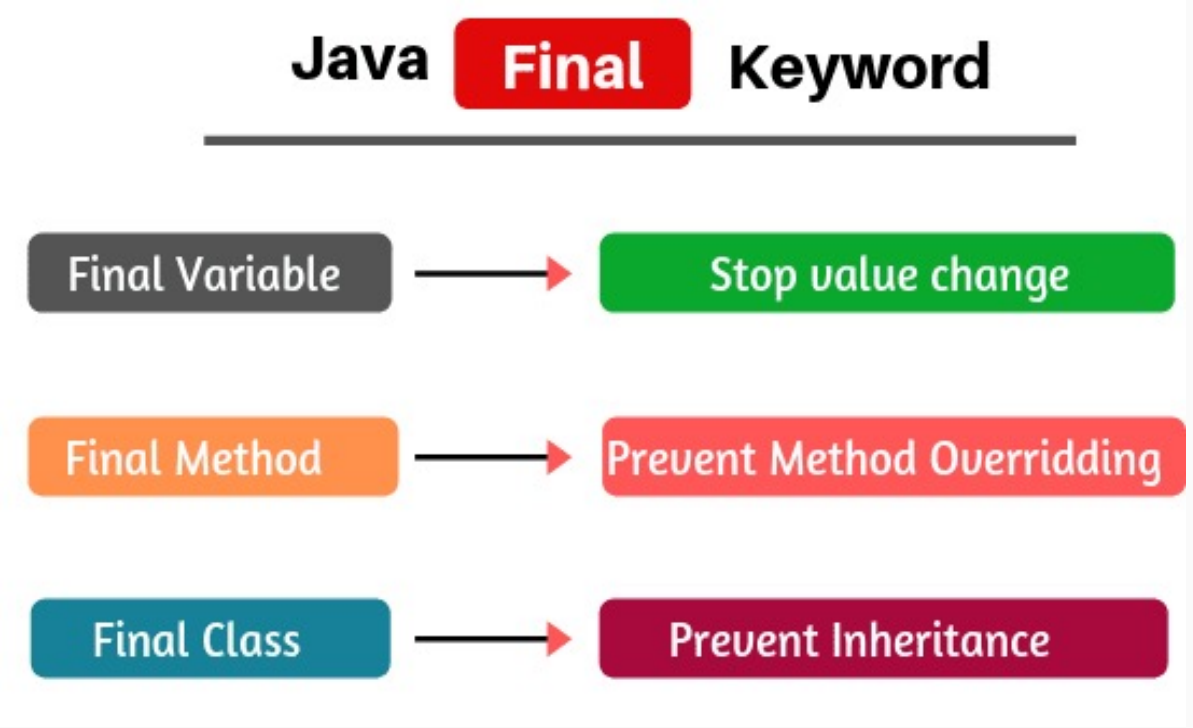
Can't Override

Final Class

Can't Inherit

Final Keyword

- Used to restrict the user.
- Makes the features unchangeable.
- Final keyword is only applicable to a variable, a method, or a class.

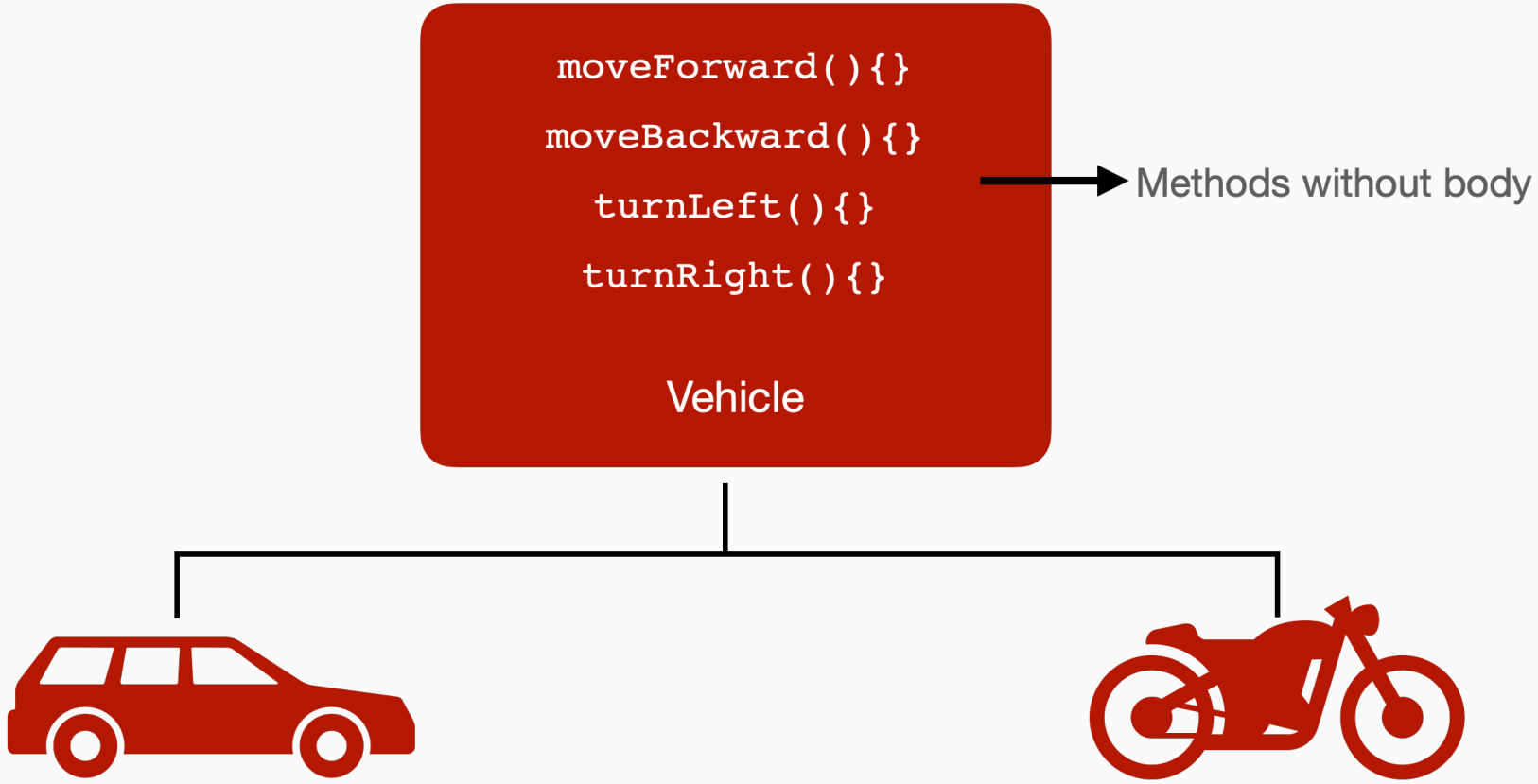


CYDEO

OOP Abstraction

What Is Abstraction?

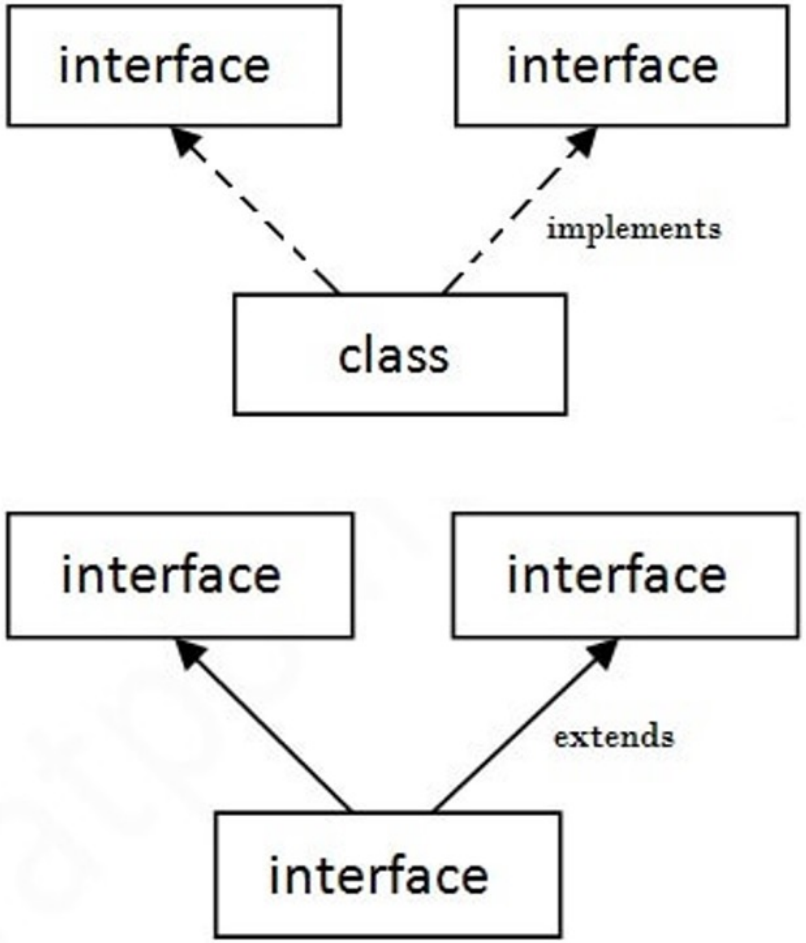
- Focus only on relevant properties of the problem
- 'Ignore' details.

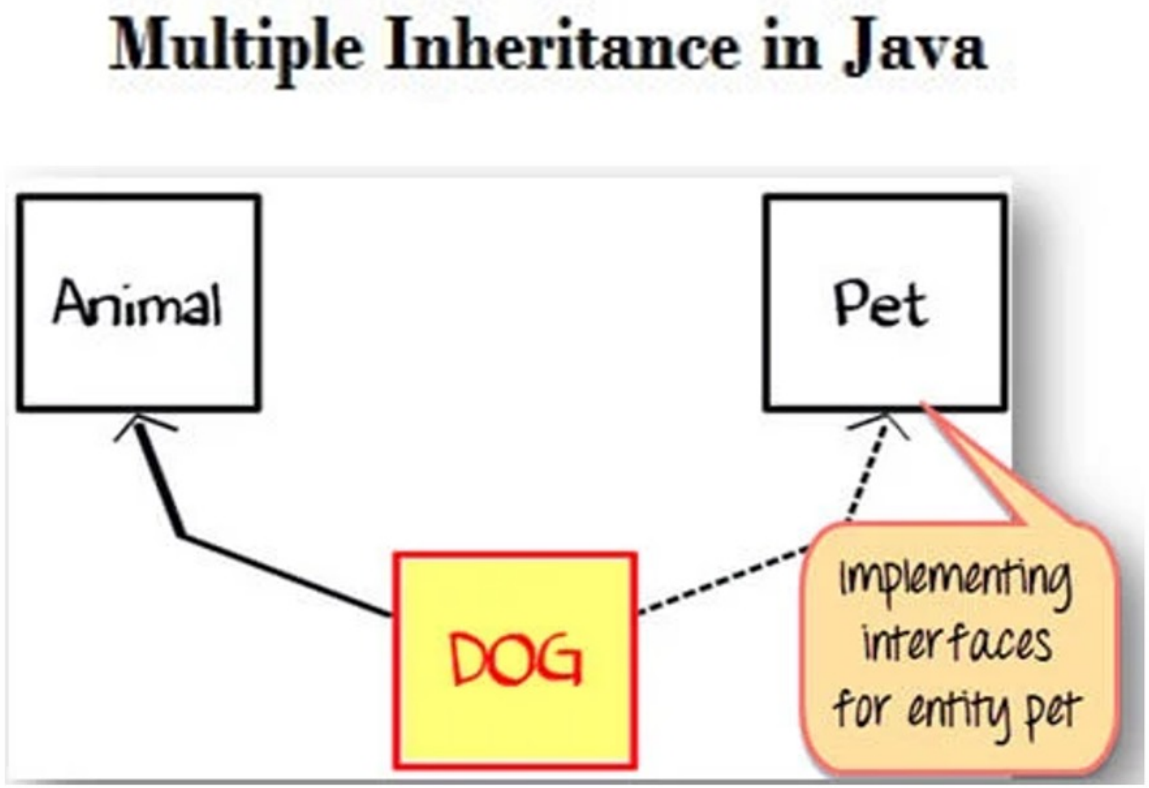


Abstraction

- 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

What Is An Interface?

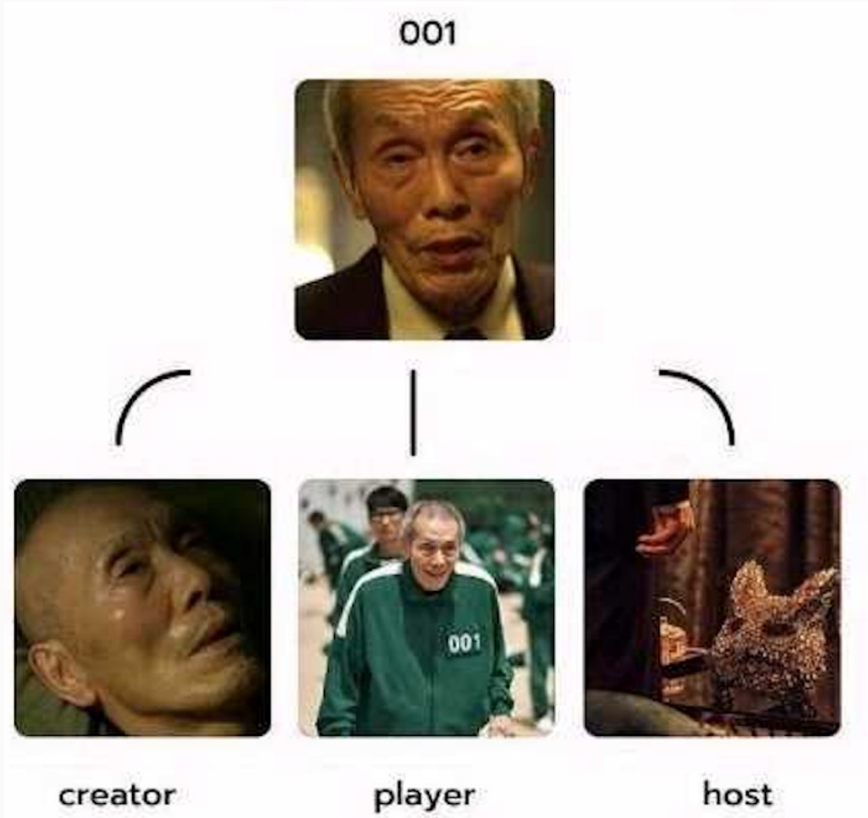




CYDEO

OOP Polymorphisn

What Is Polymorphism?



CYDEO

Poly + Morphism (Many Forms)

- Ability of the objects to take on many forms
- Occurs when reference type is parent class/interface and object type is child

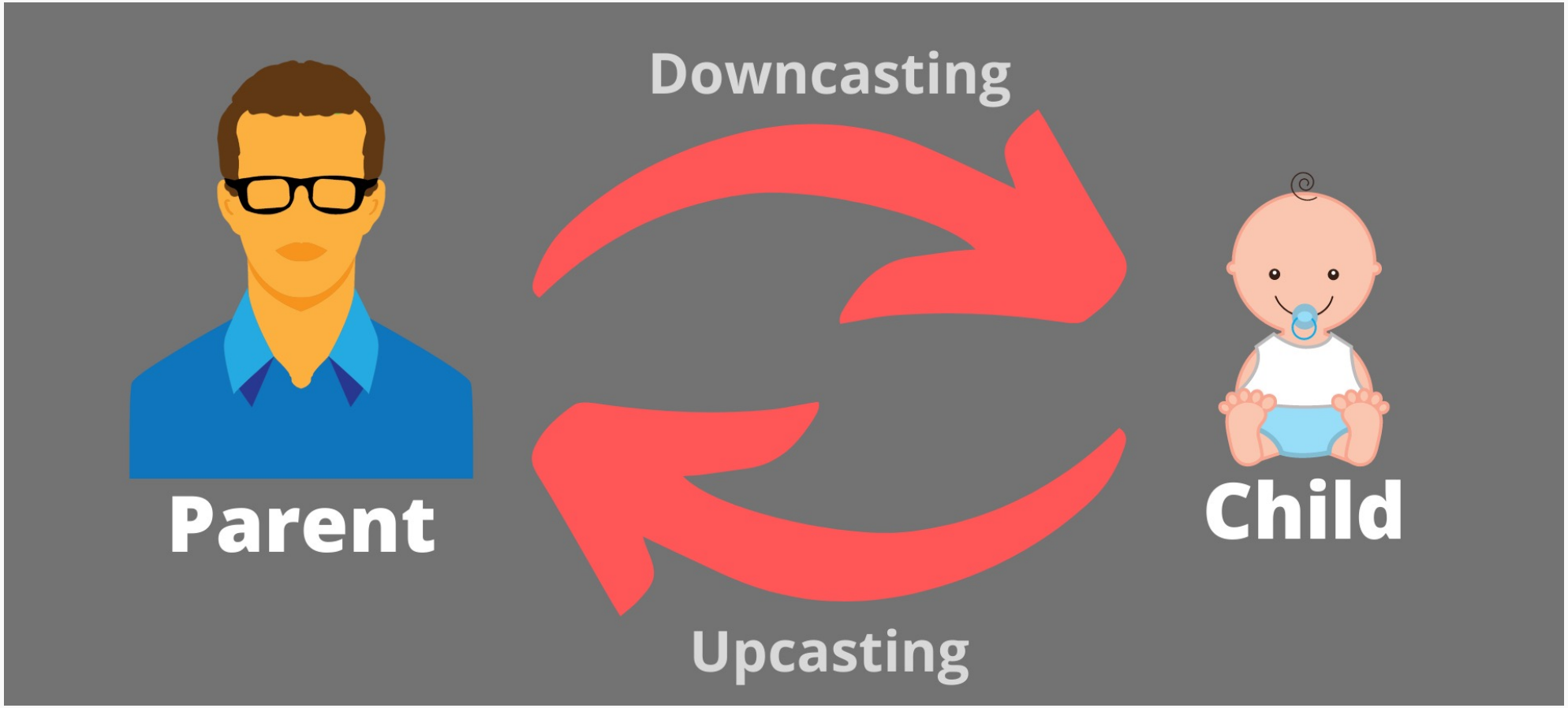
```
WebDriver driver;

driver = new FireFoxDriver();

driver = new EdgeDriver();

driver = new SafariDriver();
```

CYDEO

Reference Type Castings

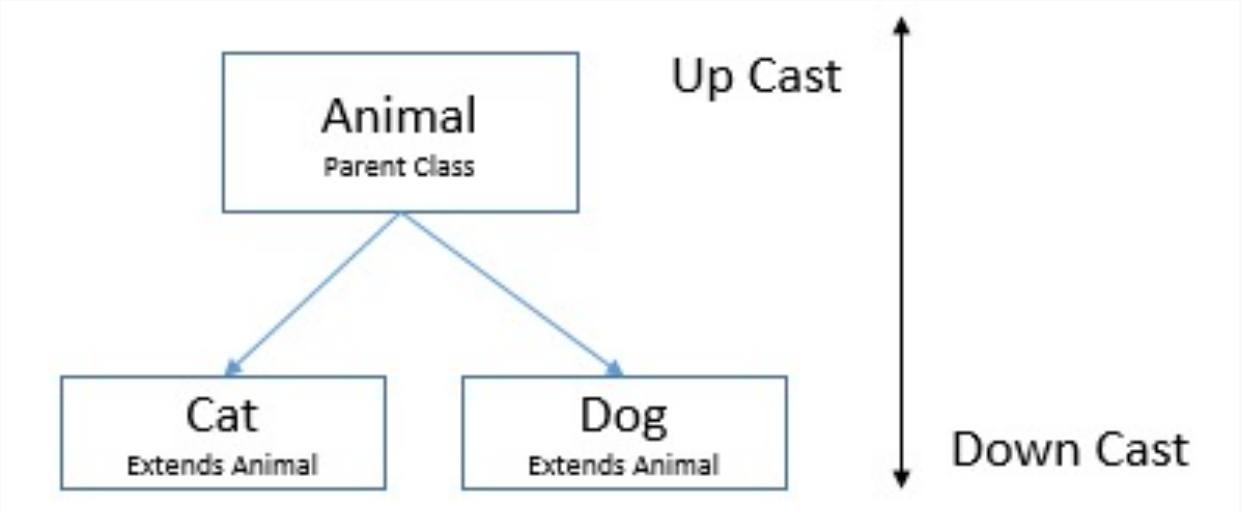
What Are Reference Type Castings?

- Casting one reference type to another



Reference Type Castings

- There must be IS A (inheritance) relation between the object type and reference type we are casting it to, otherwise ClassCastException will be thrown
- There are two types of reference type castings: upcasting and downcasting



Exceptions

What are Exceptions?

A girl is watching a video on Youtube on the computer

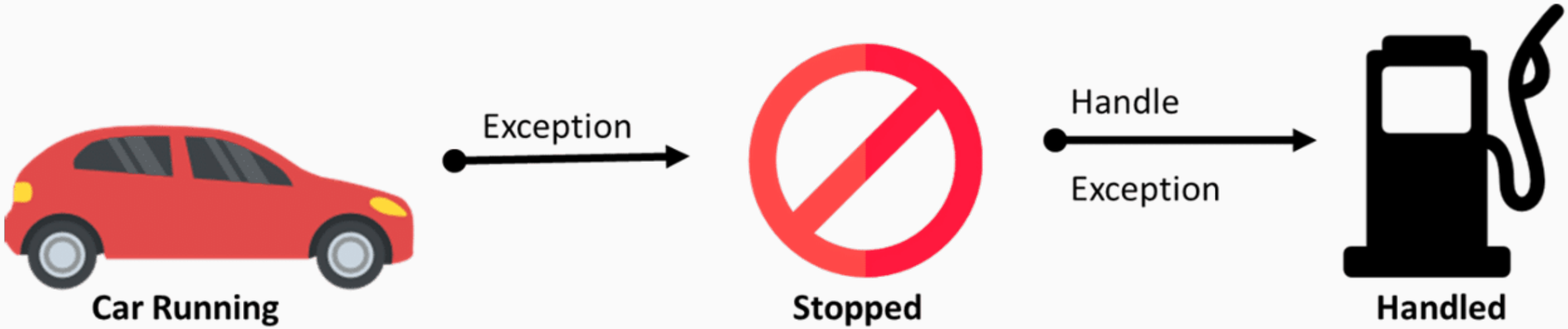
Exception

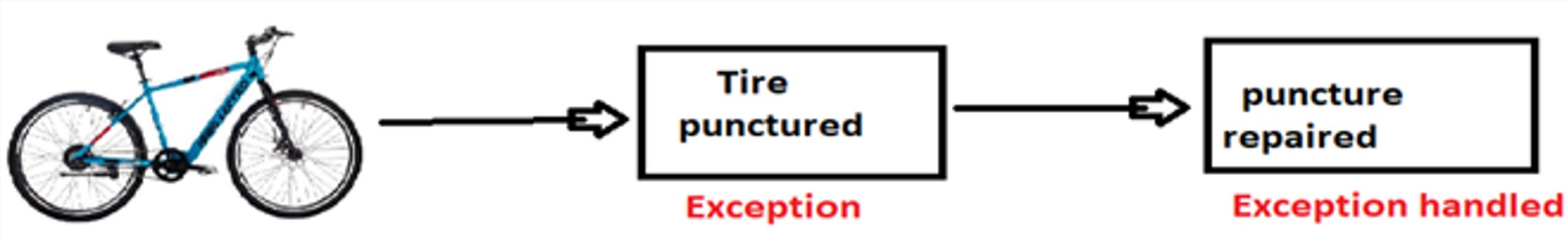
Interrupted in watching video due to internet disconnectivity suddenly

Tire punctured

Exception

Exception Handlings

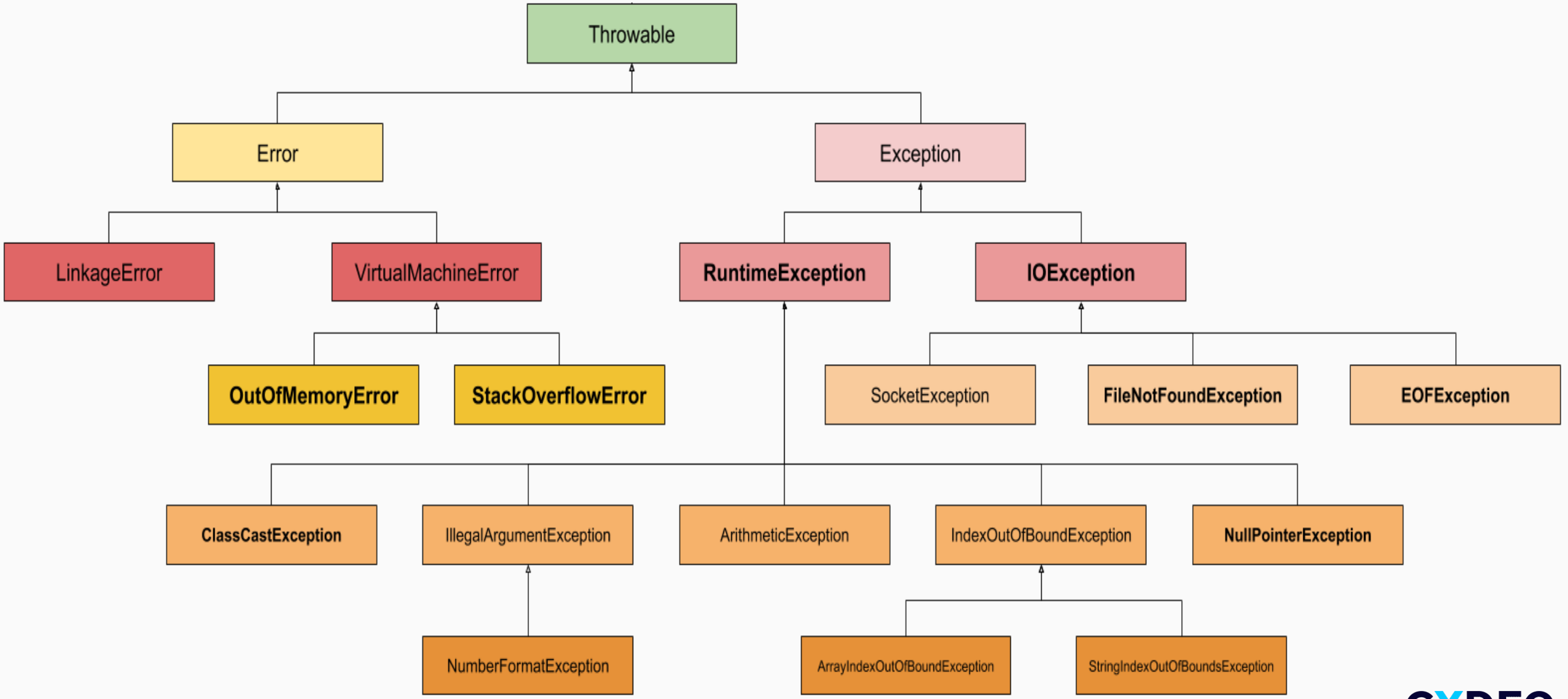




Exception

- An unwanted or unexpected event
- Occurs during the compile time or during the runtime
- There are two categories of exceptions: checked exception and unchecked exception
- To prevent exceptions from crashing our program, we must write code that detects and handles them

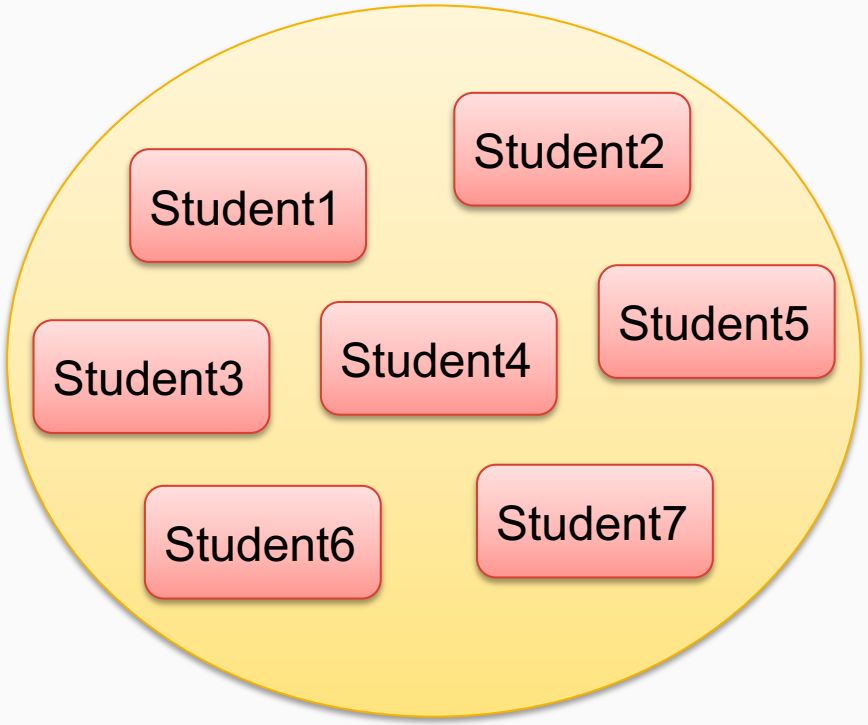
Exceptions & Errors Hierarchy

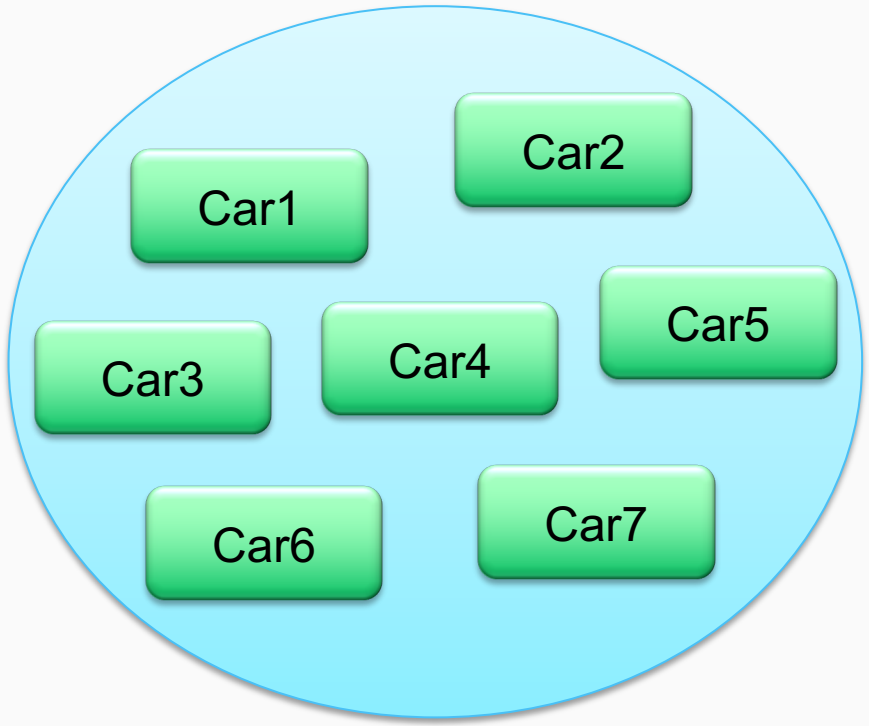


Collections

What Is Collection?

- Collection is a group of individual objects as a single entity

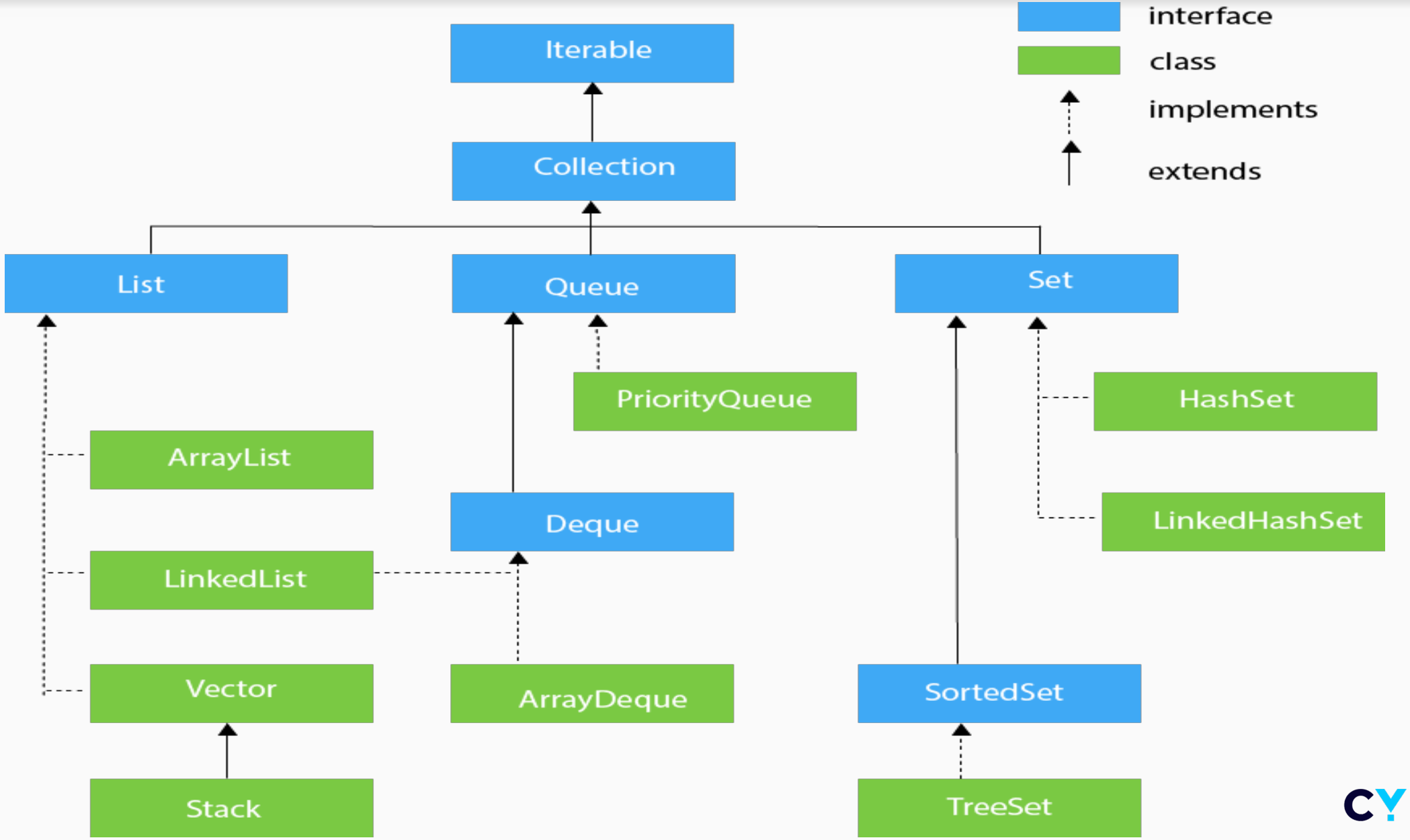




Collections

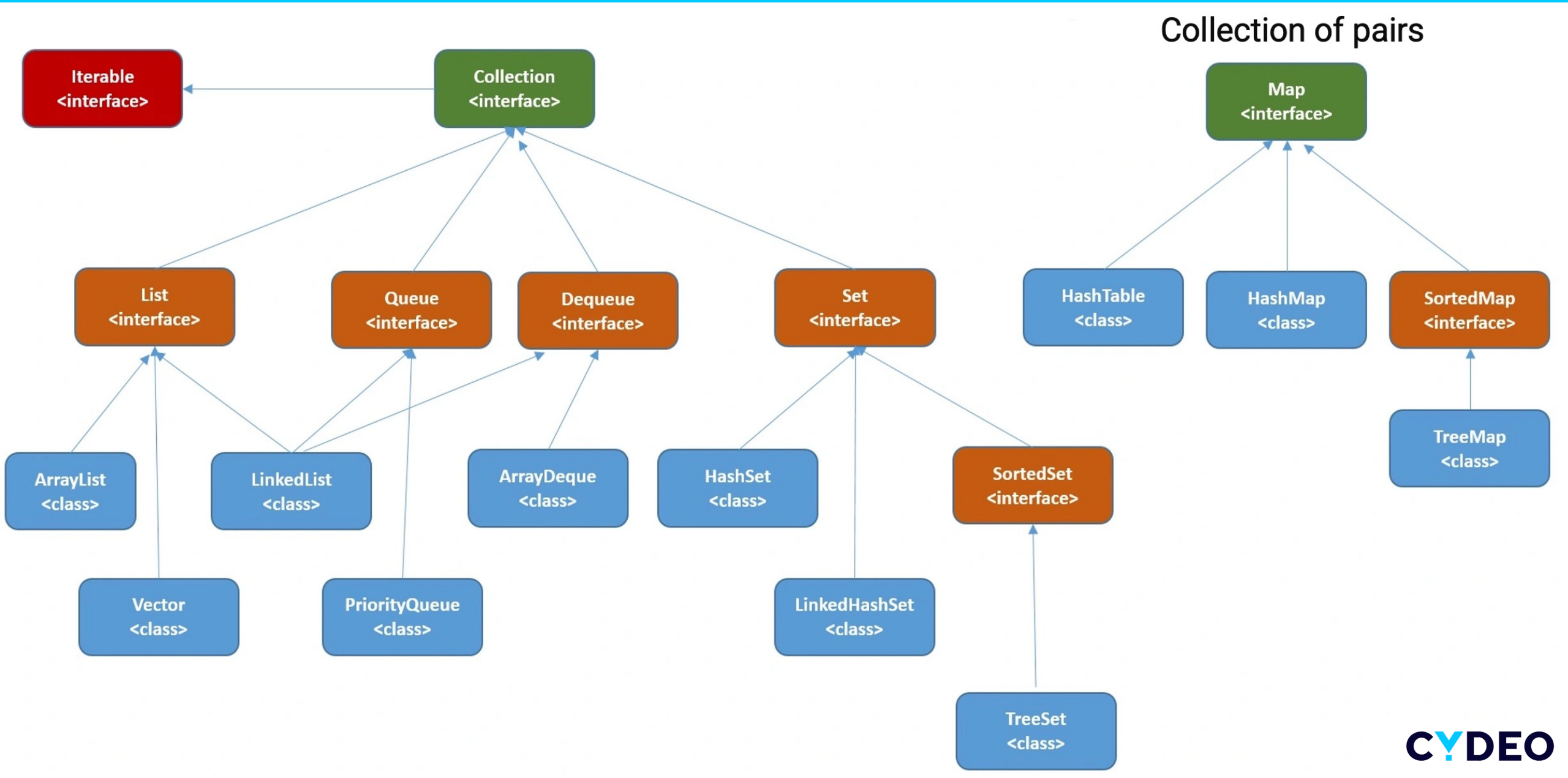
- Defines several classes and interfaces which can be used to represent a group of objects as single entity
- Growable in nature, can increase or decrease the size
- Can hold different non-primitive data types
- Standard data structure and there are ready methods to use

Collections Hierarchy



Maps

Java Collections Framework



CYDEO

Collection Of Pairs

Key	Value
name	Arthur
gender	Male
age	32
id	A01
job title	Developer
salary	110000
married	Yes