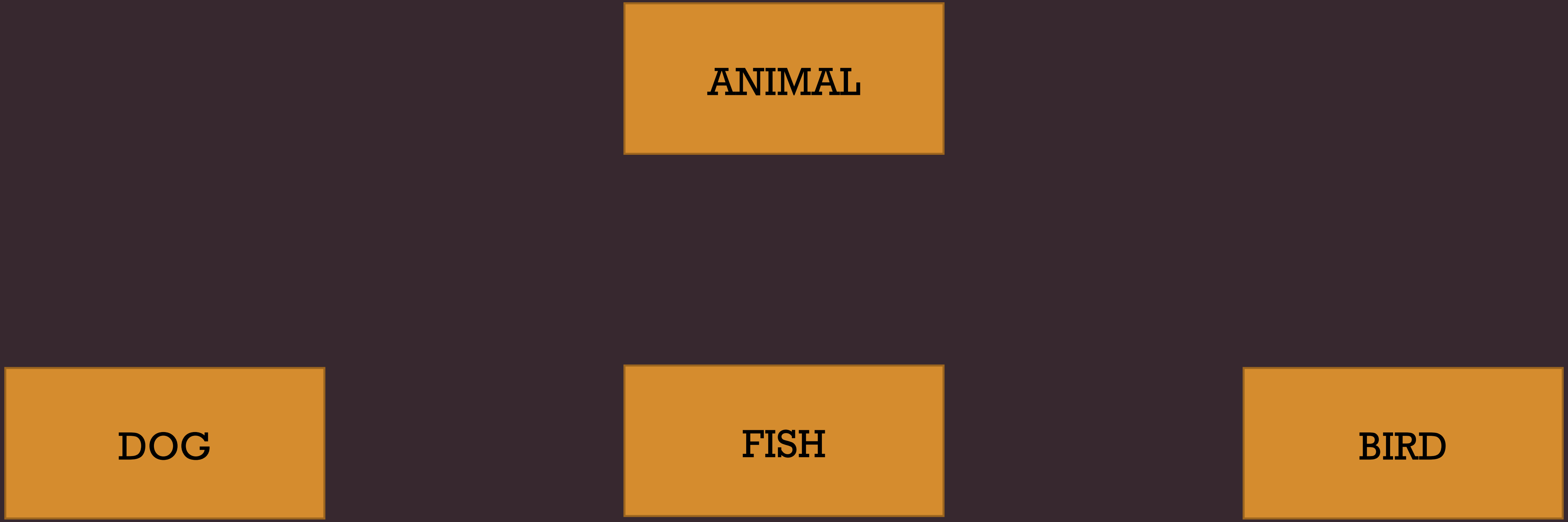
OOP PRINCIPLES

- Encapsulation
- Inheritance
- Abstraction
- Polymorphism


ANIMAL
DOG
FISH
BIRD

ANIMAL

All animals have certain characteristics.

DOG

In addition to the common animal characteristics, the dog has its own unique characteristics.

FISH

In addition to the common animal characteristics, the dog has its own unique characteristics.

BIRD

In addition to the common animal characteristics, the dog has its own unique characteristics.

ANIMAL

Name
Body
Size
Weight
Eat
Move

DOG

Name
Body
Size
Weight
Eat
Move
Legs
Tail
Teeth

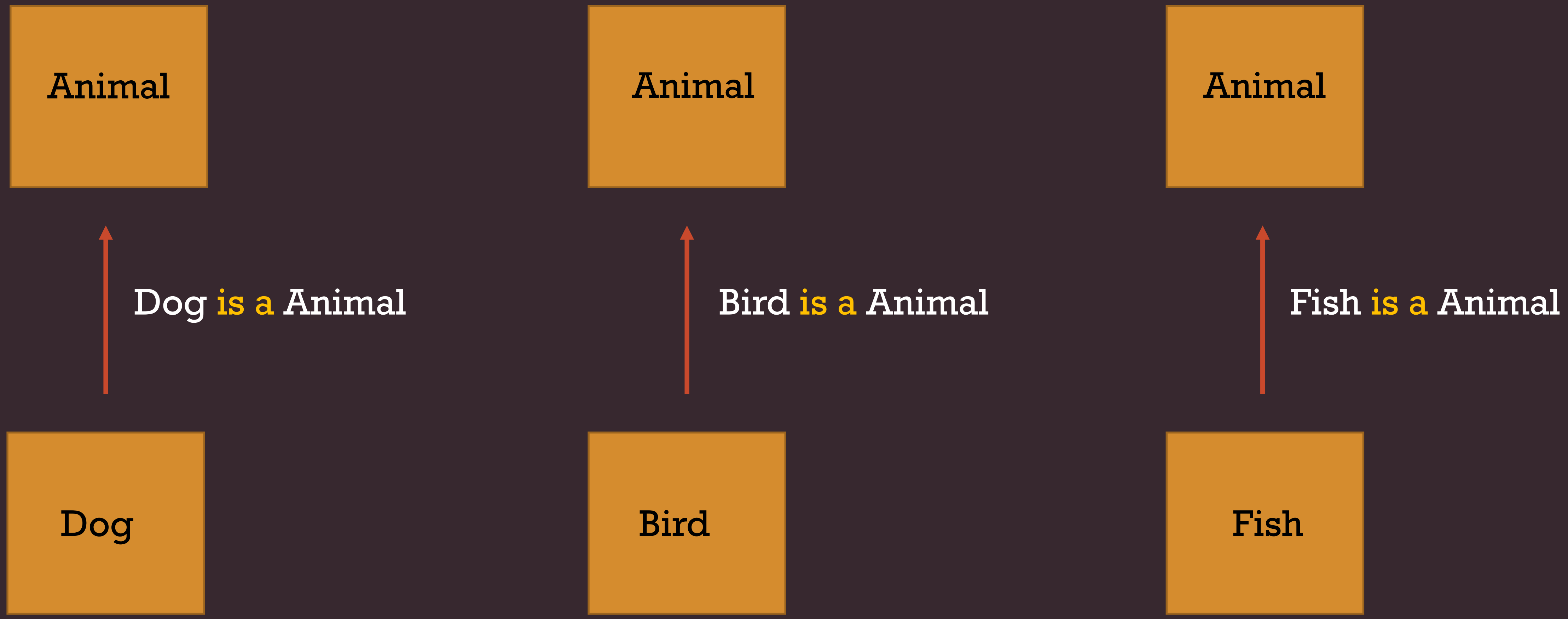
FISH

Name
Body
Size
Weight
Eat
Move
Gills
Fins
Swim

BIRD

Name
Body
Size
Weight
Eat
Move
Sing
Fly

Animal describes a very general type of creature with numerous characteristics. Because dog, fish, bird are animal, they have all the general characteristics of an animal. In addition, they have special characteristics of their own.

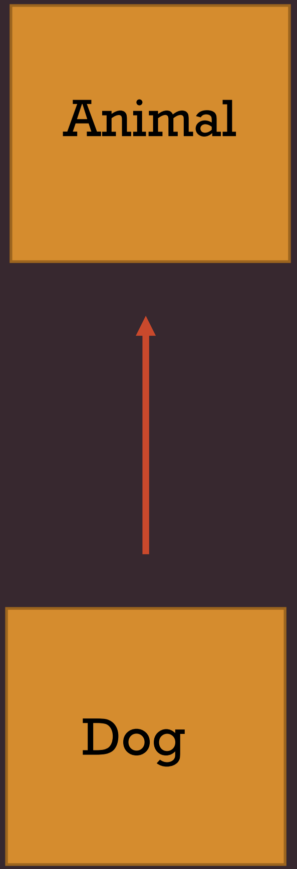

Animal
Animal
Animal
Dog is a Animal
Bird is a Animal
Fish is a Animal
Dog
Bird
Fish

Inheritance and the "IS-A" Relationship

- When an "is-a" relationship exists between objects, it means that the specialized object has all of the characteristics of the general object, plus additional characteristics that make it special.

- In OOP, inheritance is used to create an "is-a" relationship among classes.

- Inheritance is an OOP concept in Java which allows one class to inherit the features(fields and methods) of an another class.



Animal is called SUPER class and Dog is called SUB class
OR
Animal is called PARENT class and Dog is called CHILD class
OR
Animal is called BASE class and Dog is called DERIVED class

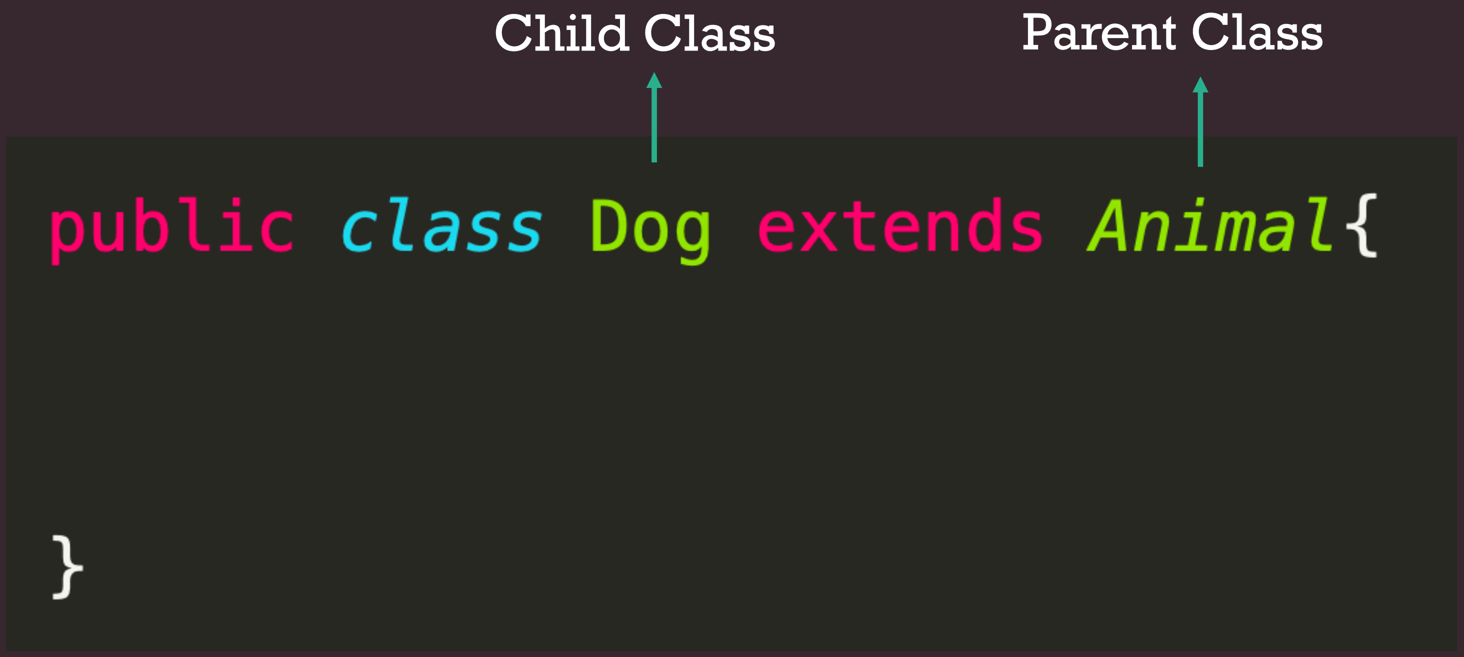
Inheritance

- Super Class: The class whose features are inherited is known as super class(or a base class or a parent class)

- Sub Class: The class that inherits the other class is known as sub class(derived class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.

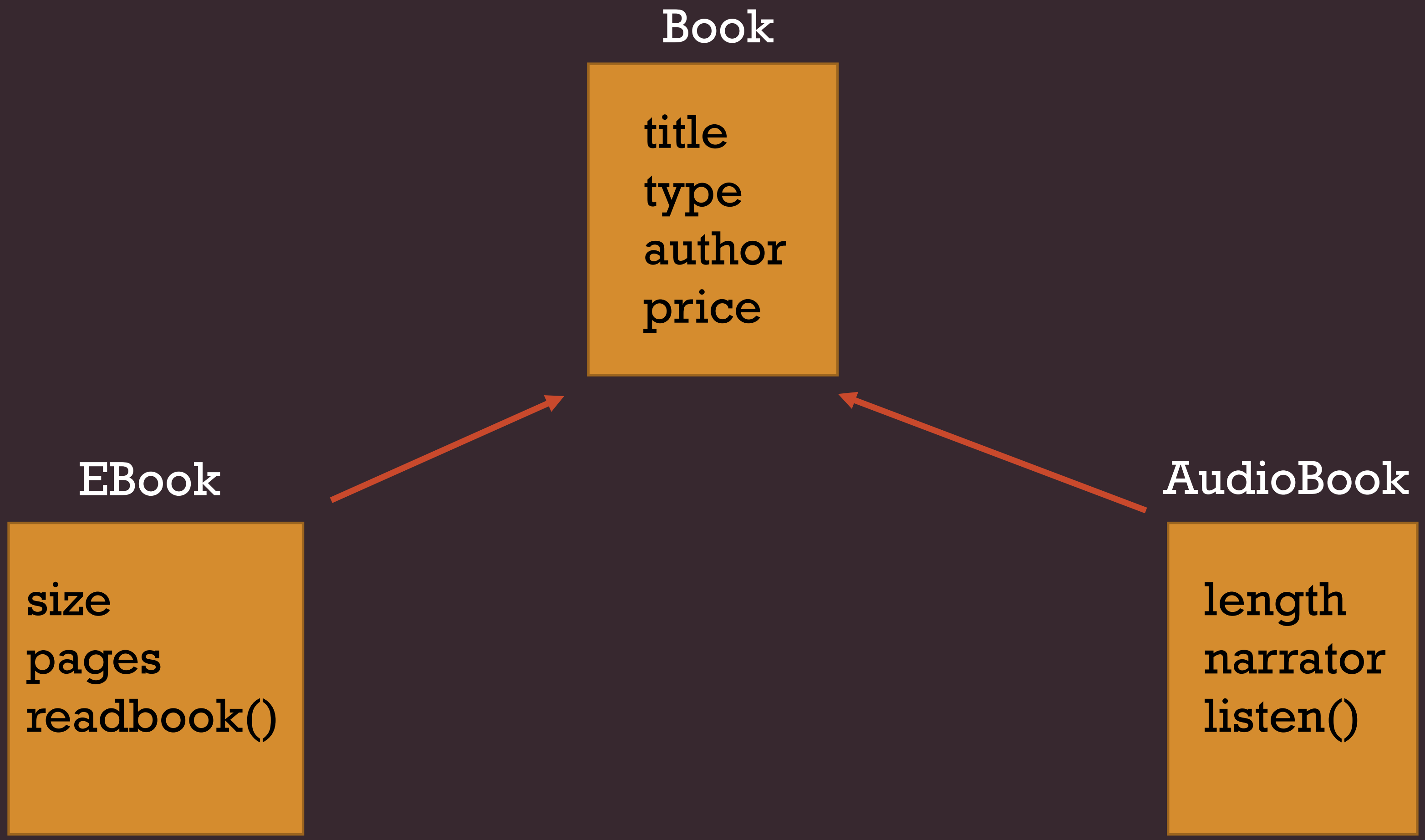
Inheritance

- The keyword used for inheritance is extends



```java
public class Dog extends Animal{


}
```

Task

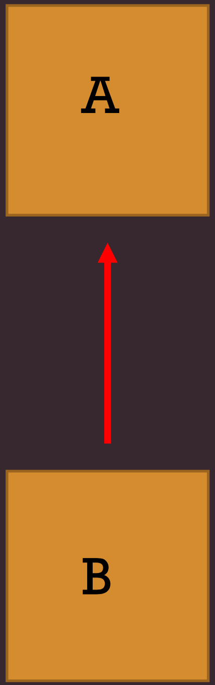

Book
title
type
author
price
EBook
size
pages
readbook()
AudioBook
length
narrator
listen()

What is inherited?

- All public variables and methods.
- All protected variables and methods.
- All default variables and methods are inherited only if super class and sub class are in the same package.
- Private variables and methods are not inherited. But it is accessible using public getter/setters.
- Constructors are not inherited.

Types of Inheritance

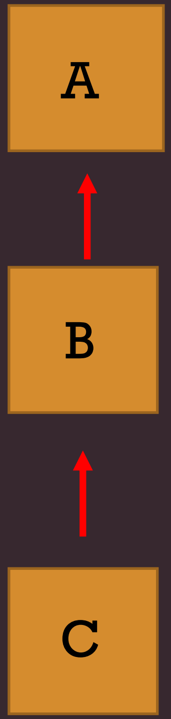
- Single Inheritance : Subclasses inherit the features of one superclass.

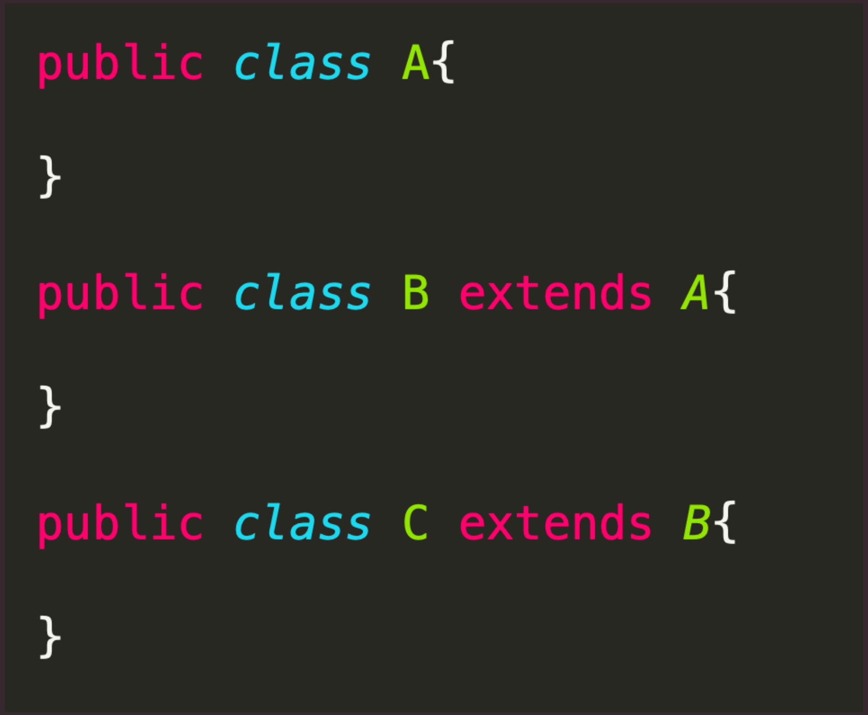


```
public class A{

}

public class B extends A{

}
```

Types of Inheritance

- Multi Level Inheritance : Subclass will be inheriting a SuperClass and as well as the subclass also act as the SuperClass to the other class.

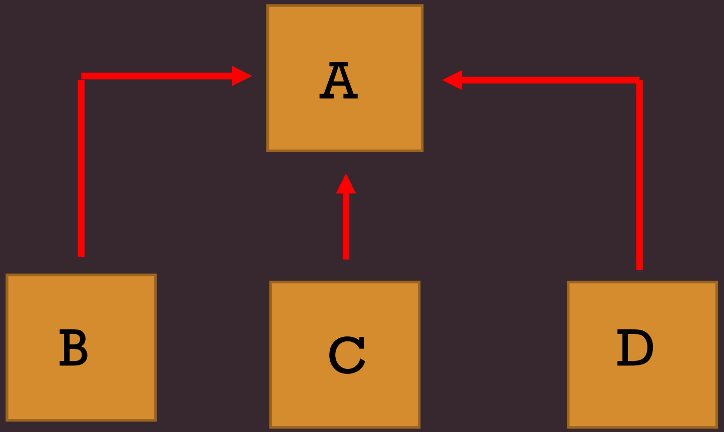




```
public class A{

}

public class B extends A{

}

public class C extends B{

}
```

Types of Inheritance

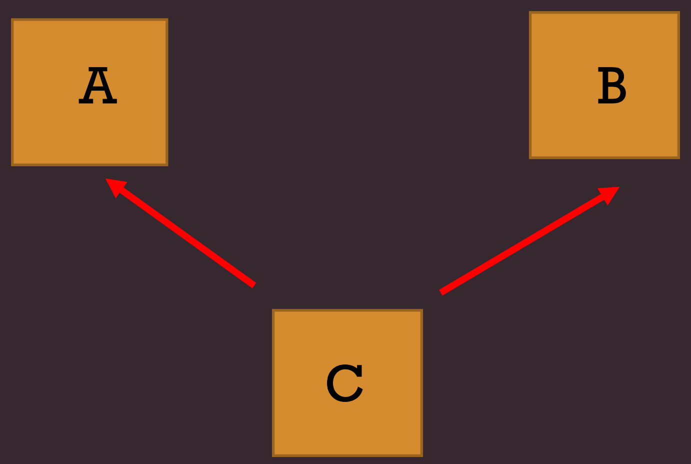
- Hierarchical Inheritance : Once class serves as superclass for more than one sub class.


$$\begin{tikzcd} & A & \\ B \arrow[ur] & C \arrow[u] & D \arrow[ul] \end{tikzcd}$$

```
public class A{

}

public class B extends A{

}

public class C extends A{

}

public class D extends A{

}
```

Types of Inheritance

- Multiple Inheritance : Java DOES NOT support multiple inheritance with classes. One class can not have more than one superclass and inherit features from all parent class.



Superclass's Constructor

- In an inheritance relationship, the superclass constructor always executes before the subclass constructor.

Calling the Super Class Constructor

- The super keyword refers to an object’s superclass. You can use the super key word to call a superclass constructor.

- If a subclass constructor does not explicitly call a superclass constructor, Java will automatically call the superclass’s default constructor, or no-arg constructor, just before the code in the subclass’s constructor executes. This is equivalent to placing the following statement at the beginning of a subclass constructor: super();

super()

- super() is used to call Parent class constructor from Child class constructor.
  - Parameters must match with parent constructor
  - It needs to be the first statement in the child class constructor
  - this() also needs to be the first statement in the constructor, so super() and this() can not be in the same constructor
  - If you do not add super() in your constructor, compiler will put one for you
  - If parent class only has constructor with parameters, then child constructor MUST make a matching super(params) call.