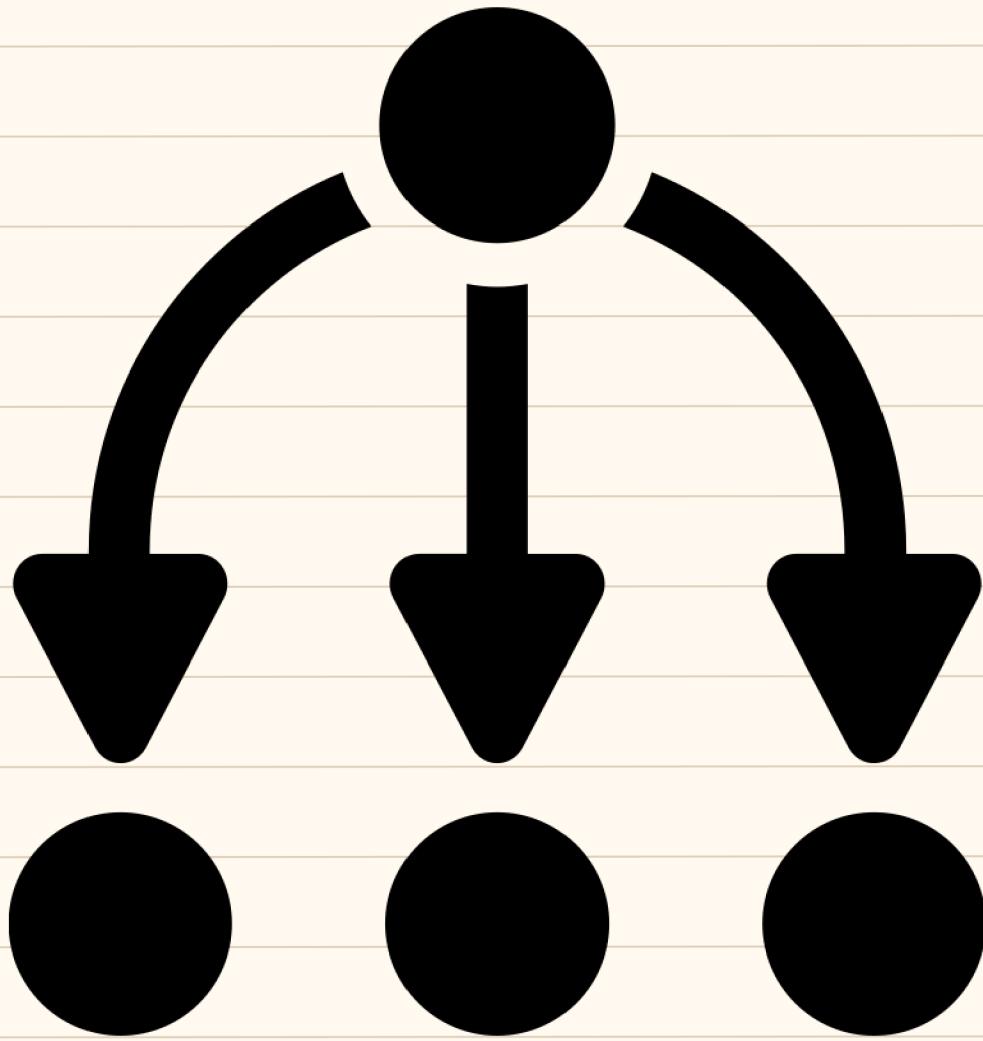


# Inheritance



Java



Muskaan(39) Neha(40)  
Nishu(41) Nitisha(42)

# Key Points

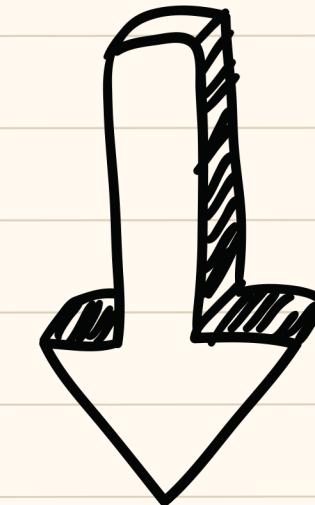
**One of the basic principals of  
Object Oriented Programming**

**Allows creation of hierarchical  
classifications**

**A general class is created that defines  
traits common to a set of related  
items.**

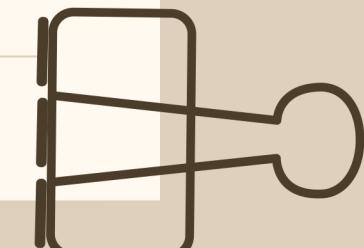
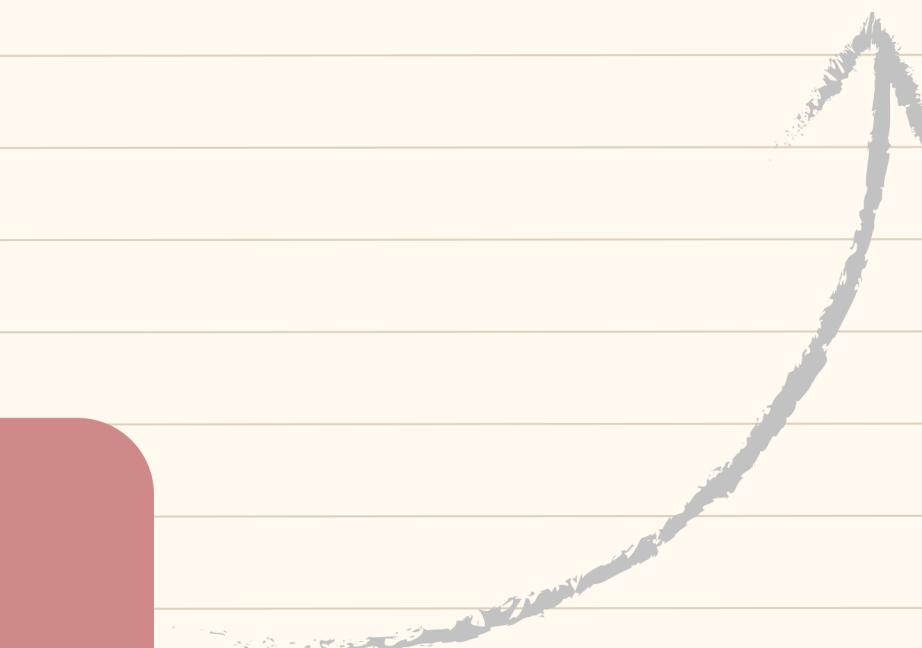
*Analogous to....*

**Parents**

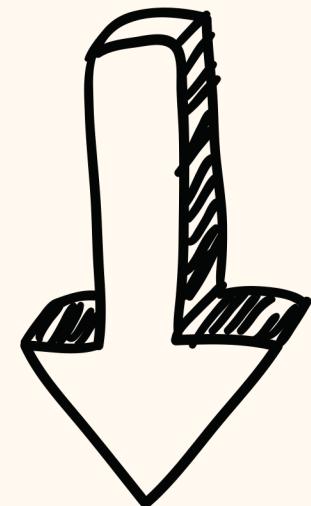


**Children**

**money, property  
traits &  
other stuffs**



**Base class**



**Child class**

Also known as



variables  
and  
methods



**Superclass**

**Subclass**



# Syntax

» Using “Extends” keyword

**class Subclass extends Superclass{**

*...definition of class*

}

```
public class Mammals { //superclass
    double age ;
    double height ;
}
class Human extends Mammals{//subclass
    String eyeColor;
}
```

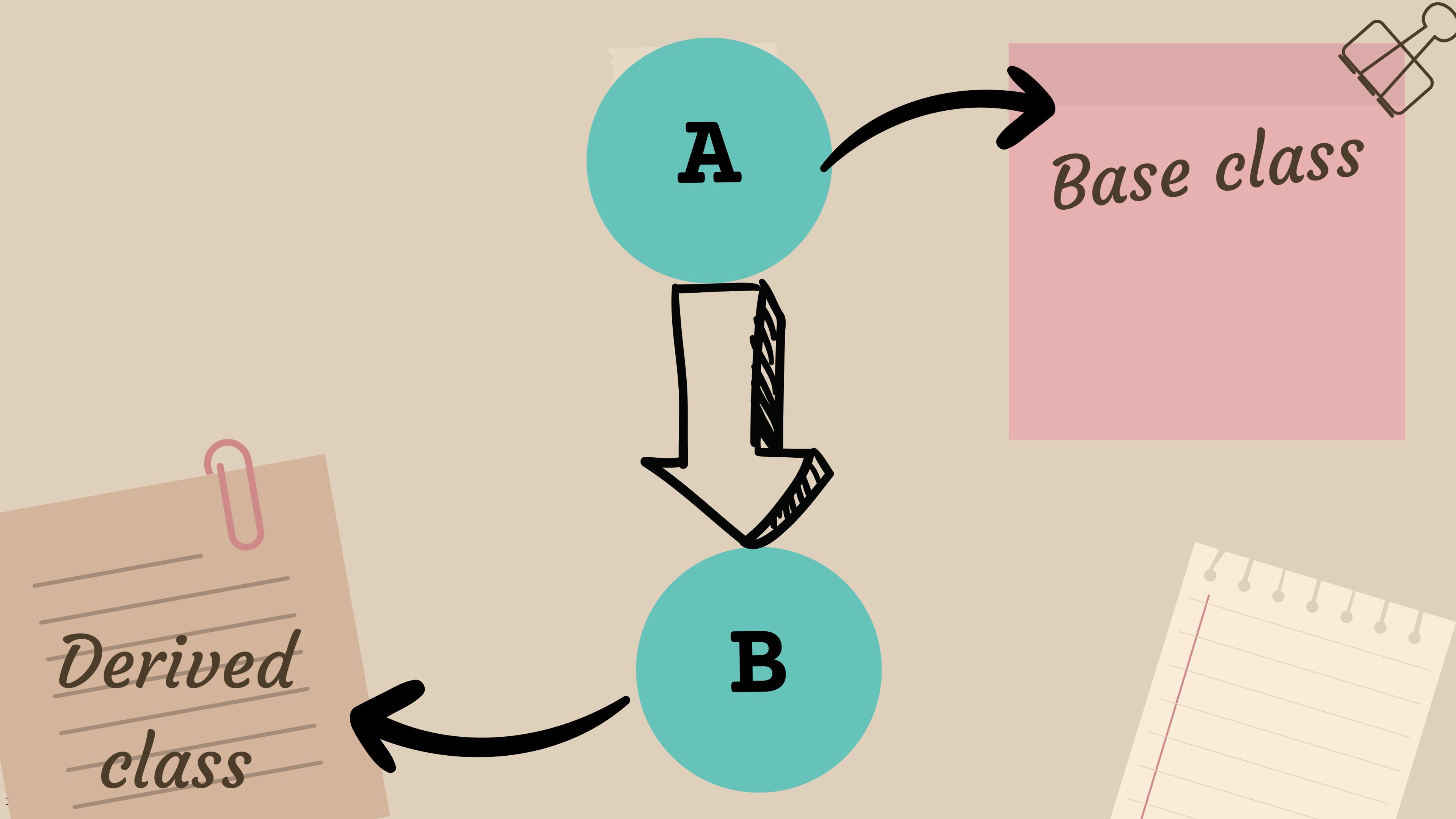
# *Types of Inheritance*

- » Single-level Inheritance
- » Multi-level Inheritance
- » Multiple Inheritance
- » Hierarchical Inheritance
- » Hybrid Inheritance

*Not allowed in JAVA*

# Single-level Inheritance

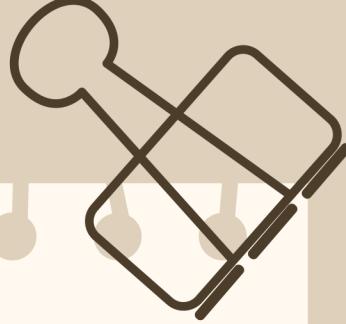
In single inheritance, subclasses inherit the features of one superclass.



# *Syntax*

```
class Superclass {  
    (fields, methods, constructors)  
}  
  
class Subclass extends Superclass {  
    (fields, methods, constructors)  
}
```

# Example



```
class Animal{
    void eat(){
        System.out.println("eating..."); }
}

class Dog extends Animal{
    void bark(){
        System.out.println("barking..."); }
}

class TestInheritance{
    public static void main(String args[]){
        Dog d=new Dog();
        d.bark();
        d.eat(); }
}
```

# Super Keyword

The 'super' keyword in Java is a reference variable that is used to refer to the immediate parent class object.

By using 'super', a subclass can invoke superclass constructors, access superclass variables and methods, and differentiate between superclass and subclass members when they have the same name.

# Super Keyword

→ **Syntax:**

**Accessing Superclass Members**

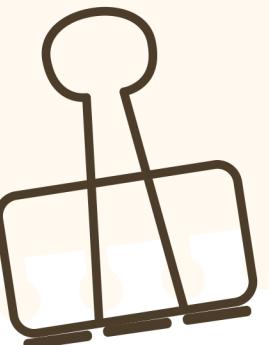
*Syntax: super.memberName*

**Invoking Superclass Constructors**

*Syntax: super(arguments)*

**Calling Superclass Methods**

*Syntax: super.methodName(arguments)*



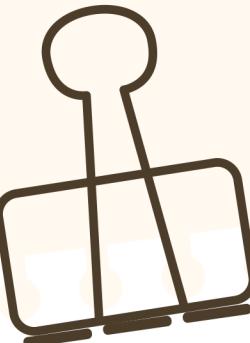
# Super Keyword

Example:

```
class A{  
    int i=10;  
}  
class B extends A{  
    int i=20;  
    void show(int i){  
        System.out.println(super.i);  
    }  
    public static void main(String[] args){  
        B ob=new B();  
        ob.show(30);  
    }  
}
```

# Super Keyword

- » Purpose of the Super Keyword
- » Accessing super class members
- » Invoking Superclass Constructors
- » Accessing Superclass Variables
- » Benefits of Using “super”



# Super Keyword



**Why do we need Super Keyword**



- ➡ To call the super class constructor.
- ➡ To access the member of super class which is hidden by sub class.

# *Use of Super Keyword*

**Super can  
be used to  
refer  
immediate  
parent  
class  
instance  
variable**

**Super can be  
used to  
invoke  
immediate  
parent class  
method**

**super() can  
be used to  
invoke  
immediate  
parent class  
constructor**

# *Parent.java*

```
class Parent {  
    String message = "Hello from Parent";  
  
    Parent() {  
        System.out.println("Parent  
constructor");  
    }  
  
    void display() {  
        System.out.println("Message from  
Parent: " + message);  
    }  
}
```



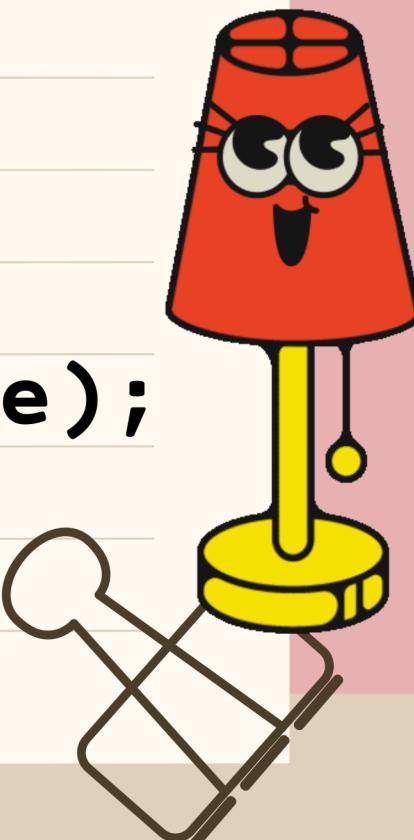
## SYNTAX



# Child.java



```
class Child extends Parent {  
    String message = "Hello from Child";  
  
    Child() {  
        super(); // invoking superclass constructor  
        System.out.println("Child constructor");  
    }  
  
    void display() {  
        super.display(); // accessing superclass method  
        System.out.println("Message from Child: "+message);  
    }  
}
```



# Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Child child = new Child();  
        child.display();  
    }  
}
```



Parent constructor

Child constructor

Message from Parent: Hello from Parent

Message from Child: Hello from Child

# *Thank you!*

**ANY QUESTIONS??**

