

# INHERITANCE

- ❑ A CLASS CAN BE A SUB-TYPE OF ANOTHER CLASS.
- ❑ THE INHERITING CLASS CONTAINS ALL THE METHODS AND FIELDS OF THE CLASS IT INHERITED FROM PLUS ANY METHODS AND FIELDS IT DEFINES.
- ❑ THE INHERITING CLASS CAN OVERRIDE THE DEFINITION OF EXISTING METHODS BY PROVIDING ITS OWN IMPLEMENTATION.

# INHERITANCE IN REAL LIFE

❑ A NEW DESIGN CREATED BY THE MODIFICATION OF AN ALREADY EXISTING DESIGN.

❑ THE NEW DESIGN CONSISTS OF ONLY THE CHANGES OR ADDITIONS FROM THE BASE DESIGN.



# INHERITANCE IN JAVA: EXTENDS

CLASS SUPERCLASS

{

}

CLASS SUBCLASS EXTENDS SUPERCLASS

{

}

# PROTECTED

❑ ATTRIBUTES AND METHODS MARKED AS

❑ **PUBLIC** ARE ALWAYS ACCESSIBLE.

❑ **PRIVATE** ARE ACCESSIBLE WITHIN THE CLASS.

❑ **PROTECTED** ARE ACCESSIBLE WITHIN THE CLASS

AND ITS SUBCLASSES



# EXAMPLE

```
CLASS BASE {  
    PROTECTED INT A,B;  
    VOID INIT()  
    {  
        A=10;    B=20;  
    } }  
CLASS DERIVED EXTENDS BASE {  
    VOID SWAP() {  
        A=A+B;  B=A-B;  A=A-B;  
        SYSTEM.OUT.PRINTLN("A : "+A+"  
        B : "+B);  
    } }
```

```
PUBLIC CLASS INHERITANCEDEMO {  
    PUBLIC STATIC VOID MAIN(STRING  
        ARGS[])  
    {  
        DERIVED OBJ=NEW DERIVED();  
        OBJ.INIT();  
        OBJ.SWAP();    }  
}
```

# OVERRIDING

THE INHERITING CLASS CAN OVERRIDE THE DEFINITION OF EXISTING METHODS BY PROVIDING ITS OWN IMPLEMENTATION.



# EXAMPLE

```
CLASS ANIMAL {  
PUBLIC VOID EAT() {  
SYSTEM.OUT.PRINTLN("GENERIC  
ANIMAL EATING  
GENERICALLY");  
}  
}  
  
CLASS HORSE EXTENDS ANIMAL {  
PUBLIC VOID EAT() {  
SYSTEM.OUT.PRINTLN("HORSE  
EATING HAY, OATS, AND HORSE  
TREATS");  
}  
}
```

ChandraSekhar(CS) Baratham

```
PUBLIC CLASS OVERRIDING {  
PUBLIC STATIC VOID  
MAIN(STRING ARGS[])  
{  
HORSE OBJ=NEW HORSE();  
OBJ.EAT();  
  
}  
}
```

**Overriding  
is based on  
Object**

# DYNAMIC METHOD DISPATCH

```
CLASS BASE{  
    VOID SHOW(){  
        SYSTEM.OUT.PRINTLN("BASE");  
    } }  
CLASS SUB1 EXTENDS BASE{  
    VOID SHOW(){  
        SYSTEM.OUT.PRINTLN("SUB1");  
    } }  
CLASS SUB2 EXTENDS BASE{  
    VOID SHOW(){  
        SYSTEM.OUT.PRINTLN("SUB2");  
    } }  
}
```

ChandraSekhar(CS) Baratam

```
CLASS MAIN{  
    PUBLIC STATIC VOID  
        MAIN(STRING ARGS[]){  
        BASE B;  
        B=NEW SUB2();  
        B.SHOW();  
    }  
}
```



# SUPER KEYWORD

- ❑ “THIS” IS A REFERENCE TO THE CURRENT OBJECT
- ❑ “SUPER” IS A REFERENCE TO THE PARENT CLASS