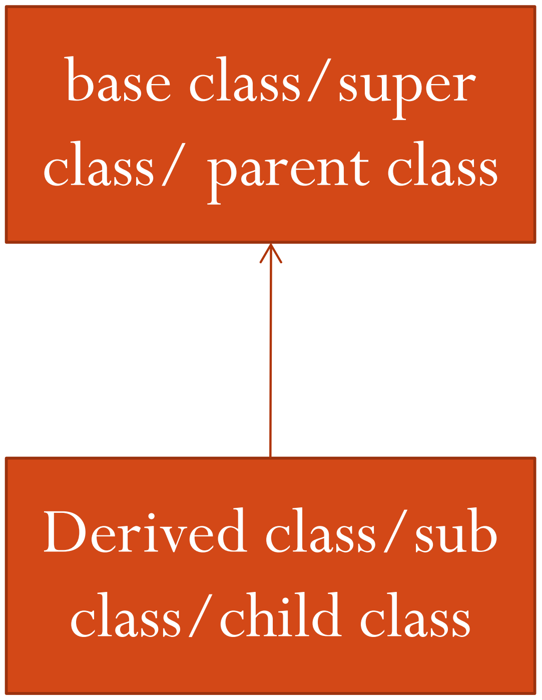
INHERITANCE

Prepared by

Dr. Rajesh Kumar Ojha
Asst. Prof., CSE, Silicon University

What is inheritance?

- It is the property by virtue of which a new class can acquire the properties of an existing class.
- The new class is known as derived class, sub class or child class and the existing class is known as base class, super class or parent class.


base class/super class/ parent class
Derived class/sub class/child class

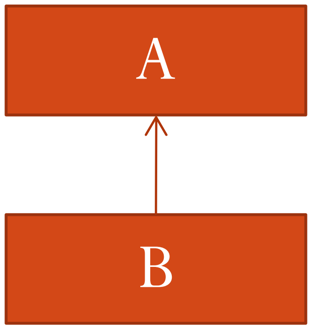
Why inheritance?

- Minimizing the amount of duplicate code in an application by sharing common code amongst several subclasses.
- Make application code more flexible to change because classes that inherit from a common super class can be used interchangeably.
- Reusability - facility to use public methods of base class without rewriting the same.
- Extensibility - extending the base class logic as per business logic of the derived class.
- Data hiding - base class can decide to keep some data private so that it cannot be altered by the derived class
- Overriding -With inheritance, we will be able to override the methods of the base class so that meaningful implementation of the base class method can be designed in the derived class.

Types of inheritance

1. Single inheritance
2. Multilevel inheritance
3. Hierarchical inheritance
4. Hybrid inheritance
5. Multilevel inheritance

Single inheritance

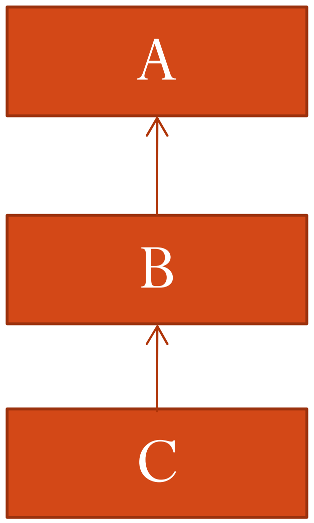


```
class A
{
   public void showA()
   {
     system.out.println("Base class show");
   }
}
class B extends A
{
   public void showB()
   {
     system.out.println("Child class show");
   }
   public static void main(String s[])
   {
     B b1 = new B();
     b1.showA(); //calling super class show
     b1.showB(); //calling local show
  }
}
```

O/P:

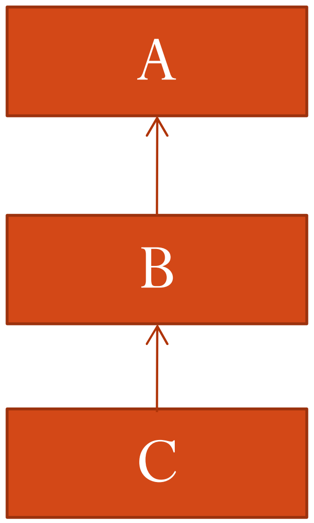
```
Base class show
Child class show
```

Multilevel inheritance



```
class A
{
   public void showA()
   {
     system.out.println("Base class showA");
   }
}
class B extends A
{
   public void showB()
   {
     system.out.println("Child class showB");
    }
}
class C extends B
{
   public void showC()
   {
     system.out.println("Child class showC");
   }
}
```

Multilevel inheritance

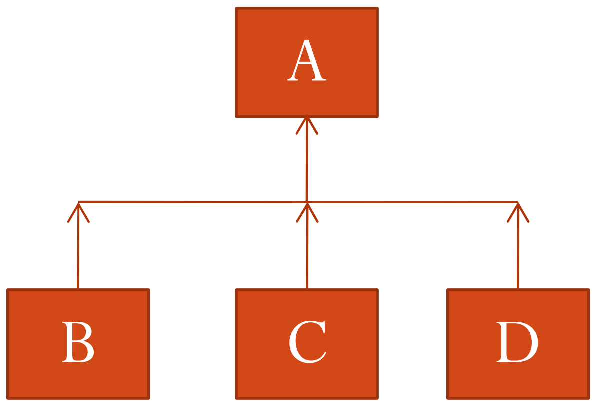


```
public static void main(String s[])
   {
     C c1 = new C();
     c1.showA(); //calling super class show
     c1.showB(); //calling super show
     c1.showC(); //calling local show
  }
}
```

O/P:

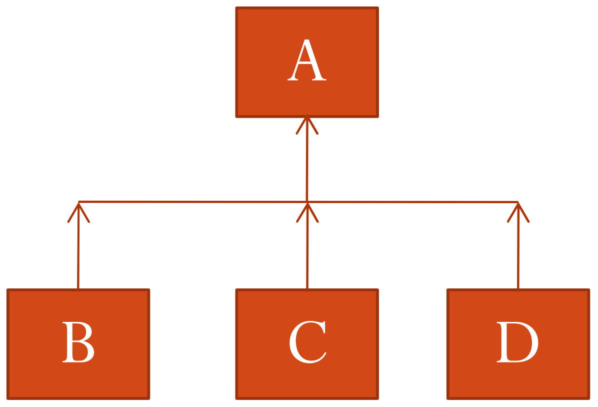
```
Base class showA
Child class showB
Child class showC
```

Hierarchical inheritance



```
class A
{
    public void showA()
    {
      system.out.println("Base class showA");
    }
}
class B extends A
{
    public void showB()
    {
      system.out.println("Child class showB");
     }
}
class C extends A
{
    public void showC()
    {
      system.out.println("Child class showC");
    }
}
```

Hierarchical inheritance

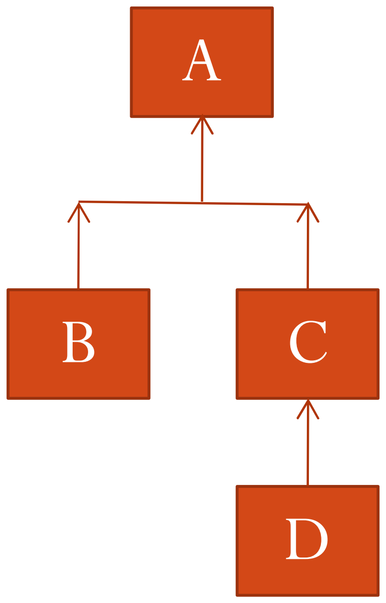


```
class D extends A
{
   public void showD()
   {
     system.out.println("Child class showD");
   }
}
public static void main(String args[])
   {
     B b1=new B();
     C c1 = new C();
     D d1 = new D();
     b1.showA();
     b1.showB();
     c1.showA();
     c1.showC();
     d1.showA();
     d1.showD();
  }
}
```

O/P:

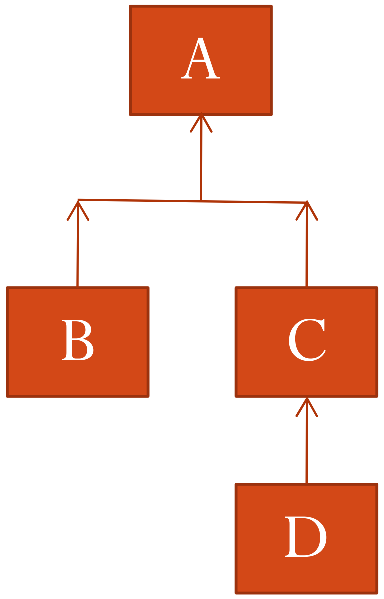
```
Base class showA
Child class showB
Base class showA
Child class showC
Base class showA
Child class showD
```

Hybrid inheritance


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

```
class A
{
   public void showA()
   {
     system.out.println("Base class showA");
   }
}
class B extends A
{
   public void showB()
   {
     system.out.println("Child class showB");
    }
}
class C extends A
{
   public void showC()
   {
     system.out.println("Child class showC");
   }
}
```

Hybrid inheritance

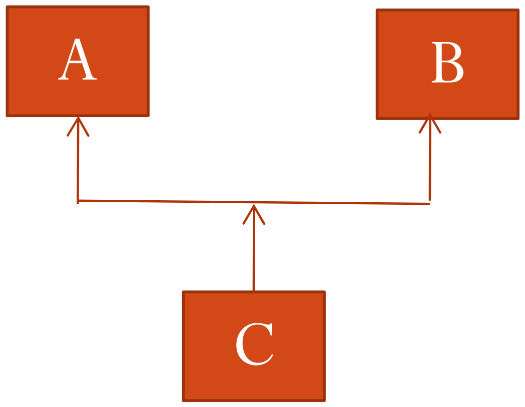


```
class D extends C
{
    public void showD()
    {
      system.out.println("Child class showD");
    }
}
public static void main(String args[])
    {
      B b1=new B();
      C c1 = new C();
      D d1 = new D();
      b1.showA();
      b1.showB();
      c1.showA();
      c1.showC();
      d1.showA();
      d1.showC();
      d1.showD();
    }
}
```

O/P:

```
Base class showA
Child class showB
Base class showA
Child class showC
Base class showA
Child class showC
Child class showD
```

Multiple inheritance



•Java does not support multiple inheritance directly with multiple classes as parents.

• According to Java in case of a multiple inheritance more than one parent can not be class rather can be interfaces.

Thank you