class Parent {  
 public void show() {  
 System.*out*.println("I'm Parent !!!");  
 }  
  
 public void parentMethod()  
 {  
 System.*out*.println("I'm a Parent Method !!");  
 }  
}  
  
class Child extends Parent {  
 public void show() {  
 System.*out*.println("I'm Child !!!");  
 }  
  
 public void childMethod() {  
 System.*out*.println("I'm child method !!!");  
 }  
}

Parent ch2 = new Child() ;

Here , in this example the

The statement ch2.childMethod(); does not work because the reference ch2 is of type Parent, even though the object it refers to is of type Child. In Java, when you use a reference of a parent class to point to an object of a child class (as in your case with Parent ch2 = new Child();), you are essentially restricting the reference to the methods and members that are available in the parent class.

This is known as reference type compatibility.

You can only call methods that are defined in the Parent class or overridden in the Child class.

If you want to access the childMethod(), you would need to cast the ch2 reference to the Child type:

((Child)ch2).childMethod(); // This works, but be cautious with type casting

**Binding :**

Connecting a method call to the method body is known as binding.

**Dynamic binding :**

When type of the object is determined at run-time, it is known as dynamic binding.

Parent ch = new Child() ;

In the above example object type cannot be determined by the compiler, because the instance of Child is also an instance of Parent.So compiler doesn't know its type, only its base type.

**Java Runtime Polymorphism with Data Member**

Only a method is overridden, not the data members by the child class.

class Bike{  
 int speedlimit=90;  
}  
  
  
class Honda extends Bike {  
 int speedlimit= 150;  
 int price = 1220000;  
}  
  
public class DataMemberPolymorphism {  
  
 public static void main(String[] args) {  
 Bike b = new Honda() ;  
 System.*out*.println(b.speedlimit);  
 *// System.out.println(b.price) ; does not work*

System.*out*.println( ((Honda)b).speedlimit); *// This works*

}  
  
}