

this pointer

```
class Student{  
    int rollno;  
    String name;  
    float fee;  
  
    Student(int rollno,String name,float fee)  
    {  
        this.rollno=rollno;  
        this.name=name;  
        this.fee=fee;  
  
        //rollno=rollno1;  
        //name=name1;  
        //fee=fee1;  
    }  
  
    void display()  
    {  
        System.out.println(rollno+" "+name+" "+fee);  
    }  
}  
  
class TestThis1 {  
    public static void main(String args[])  
    {  
        Student s1=new Student(111,"ankit",5000f);  
        Student s2=new Student(112,"sumit",6000f);  
        s1.display();  
        s2.display();  
    }  
}
```

Without using this pointer

o/p
0 null 0.0
0 null 0.0

Using this pointer

o/p

111 ankit 5000
112 sumit 6000

Using the this Keyword

Within an instance method or a constructor, *this* is a reference to the *current object* — the object whose method or constructor is being called. You can refer to any member of the current object from within an instance method or a constructor by using *this*.

Using this with a Field

The most common reason for using the *this* keyword is because a field is shadowed by a method or constructor parameter.

For example, the `Point` class was written like this

```
public class Point {  
    public int x = 0;  
    public int y = 0;  
  
    //constructor  
    public Point(int a, int b) {  
        x = a;  
        y = b;  
    }  
}
```

but it could have been written like this:

```
public class Point {  
    public int x = 0;  
    public int y = 0;  
  
    //constructor  
    public Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```