

## \*\*\* What is Variable Shadow in Java ?

If same name variables and method from variables are having exactly same name then method level variable will hide class level variable **inside the method OR constructor OR block body**. This concept is known as Variable Shadow.

Case 1 :

```
package com.sit.variable_shadow;

class Test {
    static int a = 100; //Static Field
    int b = 200; //Non static field
    public void printData(int a) //a is parameter variable
    {
        int b = 400; //Local Variable
        //System.out.println(a);
        //System.out.println(b);
    }

    public class VariableShadow {
        public static void main(String[] args) {
            Test t1 = new Test();
            t1.printData(100);
        }
    }
}
```

Case 2 :

From the above program, It is clear that inside the method body always method level variables are having more priority. If we want to represent class level fields inside the method OR block OR constructor in case of Variable hiding then we should use

- a) For Static field : We should use **class name**
- b) For non static field : We should use **this keyword**

package com.sit.variable\_shadow;

```
class Test {
    static int a = 100; //Static Field
    int b = 200; //Non static field
    public void printData(int a) //a is Parameter Variable
    {
        int b = 400; //Local Variable
        //System.out.println(a); //100
        //System.out.println(b); //200
    }

    public class VariableShadow {
        public static void main(String[] args) {
            Test t1 = new Test();
            t1.printData(100);
        }
    }
}
```

## \*\*\* What is Method Local Search algorithm in Java :

- Whenever any method, block or constructor is executed then first of all compiler will search the variable in the method, block or constructor (At Method Level).

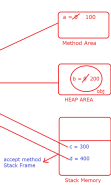
• **Variable declaration** is not available at method level then Compiler will search in the class (class level)

//Program = Diagram

```
package com.sit.variable_shadow;

class Test {
    static int a = 100; //Static Field
    int b = 200; //Non static field
    public void printData(int a) //a is parameter variable
    {
        int b = 400; //a is local variable
        //System.out.println("Static Field : "+a);
        //System.out.println("Non static Field : "+b);
        //System.out.println("Parameter Variable : "+a);
        //System.out.println("Local Variable : "+b);
    }

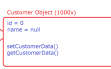
    public class MethodLocalSearchAlgo {
        void main() {
            Test t1 = new Test();
            t1.printData(100);
        }
    }
}
```



**this keyword in Java** : [1] The keyword (this) is used to refer to the current object. The this keyword is used in the following way.

\* this is a very special keyword in java which is used to refer the **current Object** as shown in the diagram.

```
public class ThisKeywordDemo {
    public static void main(String[] args) {
        Customer scott = new Customer();
        scott.getCustomerData(11, "scott");
        scott.getCustomerData();
    }
}
```



\* By using this keyword, We can refer the non-static field & non-static method anywhere in the RLC class.

\* this keyword, we **cannot** use from static context (From Static Method OR Static Block OR Static Nested Class)

\* IN JAVA, WHENEVER WE CREATE AN OBJECT THEN AUTOMATICALLY JAVA COMPILER WILL ADD THAT this to the very first parameter to all the non static method and constructor.

```
Customer java
Customer class
public class Customer {
    private int id;
    public void setData(int id) {
        this.id = id;
    }
    public void getData() {
        ID.println(id);
    }
}

public class Customer {
    private int id;
    public void setData(int Customer this, int id) {
        this.id = id;
    }
    public void getData(int Customer this) {
        ID.println(this.id);
    }
}
```

package com.sit.this\_keyword;

```
class Customer {
    private int id;
    private String name;
    public void setCustomerData(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public void getCustomerData() {
        //System.out.println("id : "+this.id);
        //System.out.println("Customer Name : "+this.name);
    }
}

public class ThisKeywordDemo {
    public static void main(String[] args) {
        Customer scott = new Customer();
        scott.setCustomerData(11, "scott");
        scott.getCustomerData();
    }
}
```