



Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

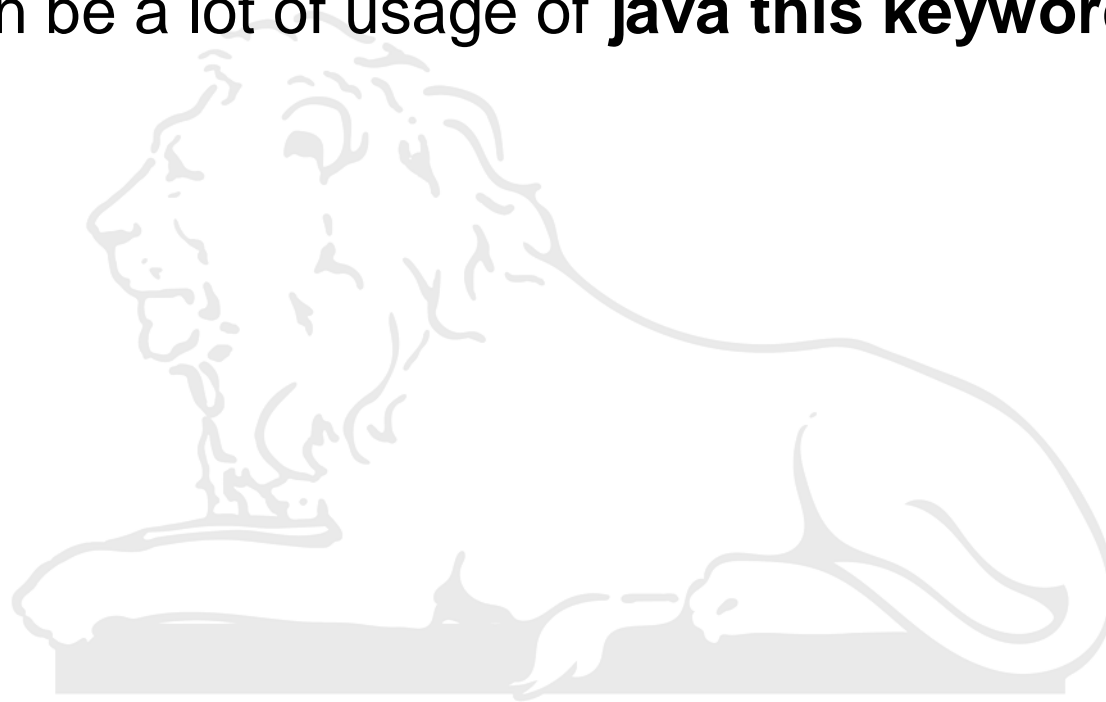
balarfcs@iitr.ac.in

<https://sites.google.com/site/balaiiitr/>



this keyword in java

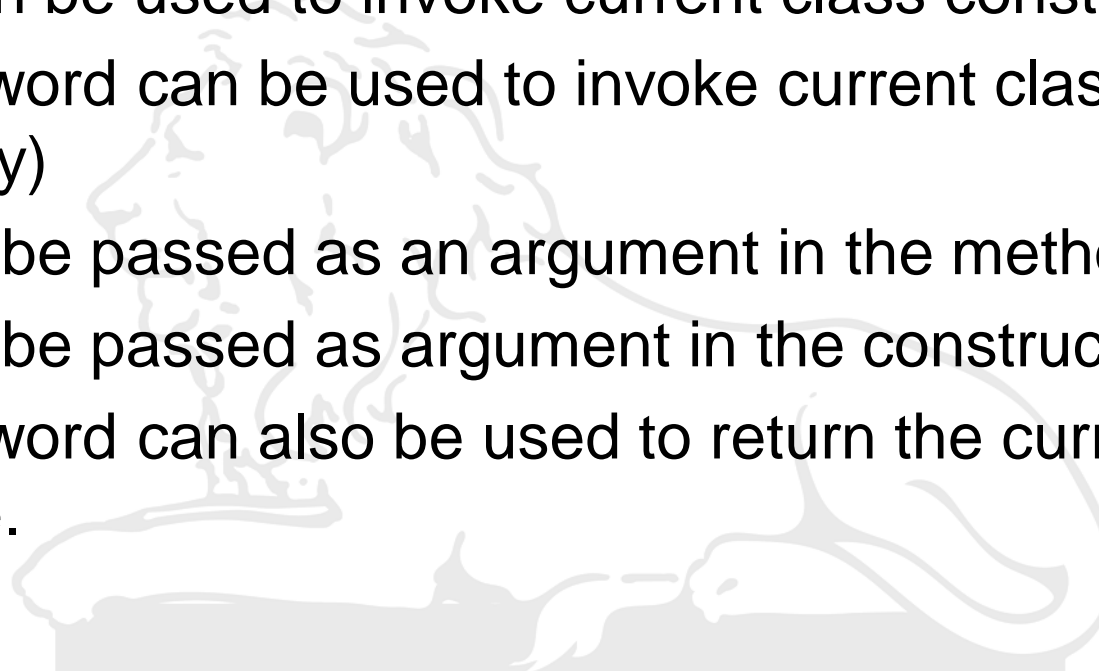
- In java, this is a **reference variable** that refers to the current object.
- There can be a lot of usage of **java this keyword**.

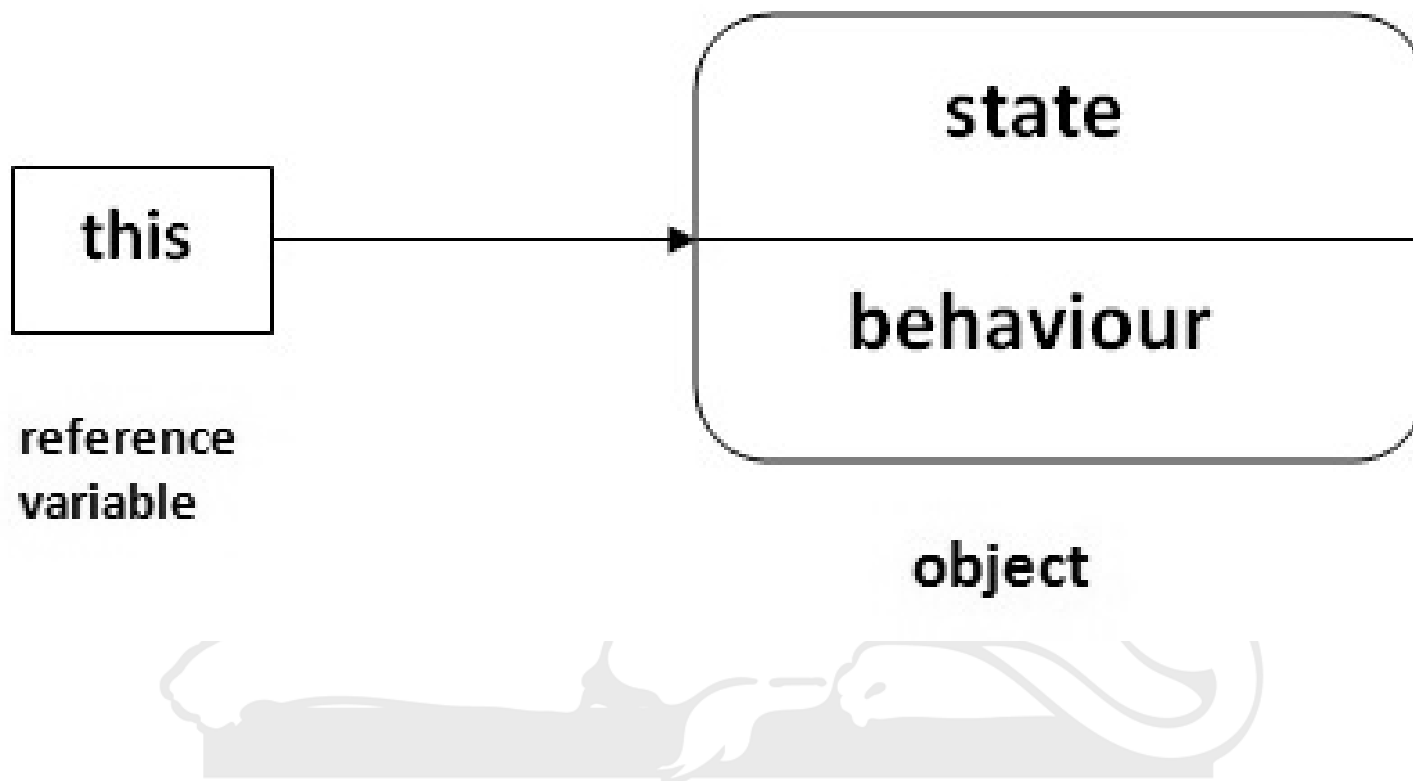




Usage of java this keyword

1. this keyword can be used to refer current class instance variable.
2. this() can be used to invoke current class constructor.
3. this keyword can be used to invoke current class method (implicitly)
4. this can be passed as an argument in the method call.
5. this can be passed as argument in the constructor call.
6. this keyword can also be used to return the current class instance.

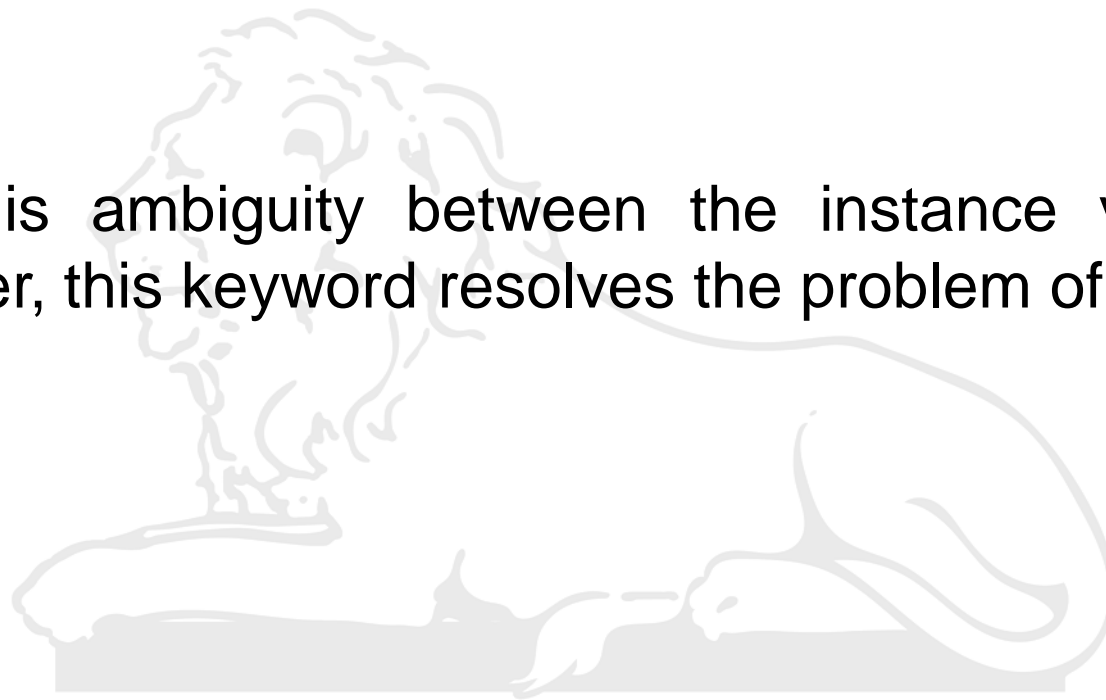




The this keyword can be used to refer current class instance variable



- If there is ambiguity between the instance variable and parameter, this keyword resolves the problem of ambiguity.





```
1 class Student10{
2     int id;
3     String name;
4     ...
5     Student10(int id,String name){
6         id = id;
7         name = name;
8     }
9     void display(){System.out.println(id+" "+name);}
10
11     public static void main(String args[]){
12         Student10 s1 = new Student10(18,"Virat");
13         Student10 s2 = new Student10(3,"Suresh");
14         s1.display();
15         s2.display();
16     }
17 }
```

Terminal

```
sh-4.3$ javac Student10.java
sh-4.3$ java Student10
0 null
0 null
sh-4.3$
```

- In the above example, parameter (formal arguments) and instance variables are same that is why we are using this keyword to distinguish between local variable and instance variable.





```
1 class Student11{
2     int id;
3     String name;
4     ...
5     Student11(int id,String name){
6         this.id = id;
7         this.name = name;
8     }
9     void display(){System.out.println(id+" "+name);}
10 public static void main(String args[]){
11     Student11 s1 = new Student11(18,"Virat");
12     Student11 s2 = new Student11(3,"Suresh");
13     s1.display();
14     s2.display();
15 }
16 }
```

Terminal

```
sh-4.3$ javac Student11.java
sh-4.3$ java Student11
18 Virat
3 Suresh
sh-4.3$
```


this() can be used to invoke current class constructor.



```
1 class Student13{
2     int id;
3     String name;
4     Student13(){System.out.println("default constructor is invoked");}
5
6     Student13(int id,String name){
7         this ();//it is used to invoke current class constructor.
8         this.id = id;
9         this.name = name;
10    }
11    void display(){System.out.println(id+" "+name);}
12
13    public static void main(String args[]){
14        Student13 e1 = new Student13(18,"Virat");
15        Student13 e2 = new Student13(3,"Suresh");
16        e1.display();
17        e2.display();
18    }
19 }
```

Terminal

```
sh-4.3$ javac Student13.java
sh-4.3$ java Student13
default constructor is invoked
default constructor is invoked
18 Virat
3 Suresh
sh-4.3$
```



Why to use this() constructor call

```
1 class Student14{
2     int id;
3     String name;
4     String city;
5     .....
6     Student14(int id,String name){
7         this.id = id;
8         this.name = name;
9     }
10    Student14(int id,String name,String city){
11        this(id,name);//now no need to initialize id and name
12        this.city=city;
13    }
14    void display(){System.out.println(id+" "+name+" "+city);}
15    .....
16    public static void main(String args[]){
17        Student14 e1 = new Student14(18,"Virat");
18        Student14 e2 = new Student14(3,"Suresh","Muradnagar");
19        e1.display();
20        e2.display();
21    }
22 }
```

 Terminal

```
sh-4.3$ javac Student14.java
sh-4.3$ java Student14
18 Virat null
3 Suresh Muradnagar
sh-4.3$
```



```
1 class Student14{
2     int id;
3     String name;
4     String city;
5     ...
6     Student14(int id,String name){
7         this.id = id;
8         this.name = name;
9     }
10    Student14(int id,String name,String city){
11        // this(id,name);//now no need to initialize id and name
12        this.city=city;
13    }
14    void display(){System.out.println(id+" "+name+" "+city);}
15    ...
16    public static void main(String args[]){
17        Student14 e1 = new Student14(18,"Virat");
18        Student14 e2 = new Student14(3,"Suresh","Muradnagar");
19        e1.display();
20        e2.display();
21    }
22 }
```

Terminal

```
sh-4.3$ javac Student14.java
sh-4.3$ java Student14
18 Virat null
0 null Muradnagar
sh-4.3$
```



```
1 class Student13{
2     int id;
3     String name;
4     Student13(){System.out.println("default constructor is invoked");}
5     .....
6     Student13(int id,String name){
7         //this ();//it is used to invoke current class constructor.
8         this.id = id;
9         this.name = name;
10        this ();
11    }
12    void display(){System.out.println(id+" "+name);}
13    .....
14    public static void main(String args[]){
15        Student13 e1 = new Student13(18,"Virat");
16        Student13 e2 = new Student13(3,"Suresh");
17        e1.display();
18        e2.display();
19    }
20 }
```

Terminal

```
sh-4.3$ javac Student13.java
Student13.java:10: error: call to this must be first statement in constructor
    this ();
      ^
1 error
```



Terminal

```
sh-4.3$ javac Student13.java
Student13.java:10: error: call to this must be first statement in constructor
    this ();
        ^
1 error
```





The this keyword can be used to invoke current class method (implicitly).

```
1 class S{
2     void m(){
3         System.out.println("Use of this keyword in JAVA");
4     }
5     void n(){
6         this.m();//no need because compiler does it for you.
7     }
8     void p(){
9         n();//compiler will add this to invoke n() method as this.n()
10    }
11    public static void main(String args[]){
12        S s1 = new S();
13        s1.p();
14    }
15 }
```

Terminal

```
sh-4.3$ javac S.java
sh-4.3$ java S
Use of this keyword in JAVA
sh-4.3$
```



this keyword can be passed as an argument in the method

```
1 class S2{
2     void m(S2 obj){
3         System.out.println("OOP-CSN-103");
4     }
5     void p(){
6         m(this);
7     }
8
9     public static void main(String args[]){
10        S2 s1 = new S2();
11        s1.p();
12    }
13 }
```

Terminal

```
sh-4.3$ javac S2.java
sh-4.3$ java S2
OOP-CSN-103
sh-4.3$
```

Reference object and this, output of both are same



```
1 class A5{
2 void m(){
3 System.out.println(this);//prints same reference ID
4 }
5
6 public static void main(String args[]){
7 A5 obj=new A5();
8 System.out.println(obj);//prints the reference ID
9
10 obj.m();
11 }
12 }
```

Terminal

```
sh-4.3$ javac A5.java
sh-4.3$ java A5
A5@659e0bfd
A5@659e0bfd
sh-4.3$
```
