Understanding 'this' Keyword:

We can use 'this' keyword in below mentioned '4' scenerios

- ⇨ It is used to resolve the **ambguity** between Instance Variables & Local Variables.
- ⇨ It is used to call present class methods.
- ⇨ It is used to return the Instance of the present class
- ⇨

this	this()
1) It is a Keyword	1) It is a Constructor call
2) It is used to call present class variables & Methods	2) It is used to call present class Constructors.
3) We cant use 'this' keyword inside a static area.	3) We need to use this() constructor call **ONLY** inside a constructor, that too as a **FIRST** statement. If we are using this() anywhere else we will be getting **C.E**

Understanding 'this' Keyword:

We can use 'this' keyword in below mentioned '4' scenerios

- ⇨ It is used to resolve the ambguity between Instance Variables & Local Variables.

Ambiguity means confusion

```
1  package com.pack1;
2
3  public class ClassA
4  {
5      int x=10;
6
7      void meth1()
8      {
9          System.out.println("meth1() called");
10         System.out.println("Instance Variable : "+x);
11     }
12     public static void main(String[] args)
13     {
14         ClassA aobj=new ClassA();
15         aobj.meth1();
16     }
17 }
```

```
<terminated> ClassA [Java Application] C:\Program
meth1() called
Instance Variable : 10
```

```java
package com.pack1;

public class ClassA
{
    int x=10;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        System.out.println("Instance Variable : "+x);
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```

```
<terminated> ClassA [Java Application] C:\Program F
meth1() called
Instance Variable : 111
```

```java
package com.pack1;

public class ClassA
{
    int x=10;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        System.out.println("Instance Variable : "+new ClassA().x);
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```

```
<terminated> ClassA [Java Application] C:\Program
meth1() called
Instance Variable : 10
```

In the above 2 copies of instance variable are there

'this' with refers to already created object in your class

```java
package com.pack1;

public class ClassA
{
    int x=10;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        System.out.println("Instance Variable : "+this.x);
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```

```
<terminated> ClassA [Java Application] C:\Program F
meth1() called
Instance Variable : 10
```

```java
package com.pack1;

public class ClassA
{
    int x=10;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        System.out.println("Instance Variable : "+this.x);
        System.out.println(this.hashCode());
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
        System.out.println(aobj.hashCode());
    }
}
```

```
<terminated> ClassA [Java Application] C:\Program
meth1() called
Instance Variable : 10
317574433
317574433
```

```java
package com.pack1;

public class ClassA
{
    int x=10;
    static int y=20;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        int y=222;
        System.out.println("Instance Variable : "+this.x);
        System.out.println("Static Variable : "+y);

    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();

    }
}
```

```
<terminated> ClassA [Java Application] C:\Program
meth1() called
Instance Variable : 10
Static Variable : 222
```

```java
package com.pack1;

public class ClassA
{
    int x=10;
    static int y=20;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        int y=222;
        System.out.println("Instance Variable : "+this.x);
        System.out.println("Static Variable : "+this.y);

    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();

    }
}
```

```
<terminated> ClassA [Java Application] C:\Program
meth1() called
Instance Variable : 10
Static Variable : 20
```

```java
package com.pack1;

public class ClassA
{
    int x=10;
    static int y=20;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        int y=222;
        System.out.println("Instance Variable : "+this.x);
        System.out.println("Static Variable : "+ClassA.y);

    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();

    }
}
```

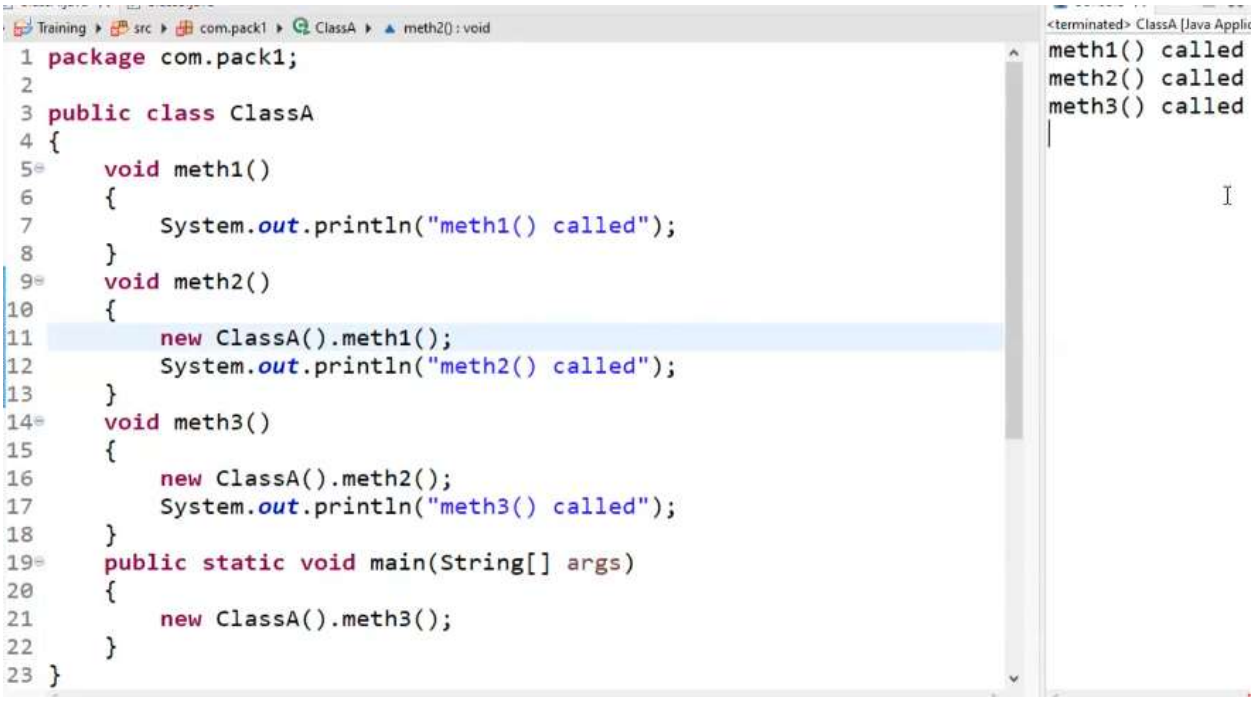
```
<terminated> ClassA [Java Application] C:\Program
meth1() called
Instance Variable : 10
Static Variable : 20
```

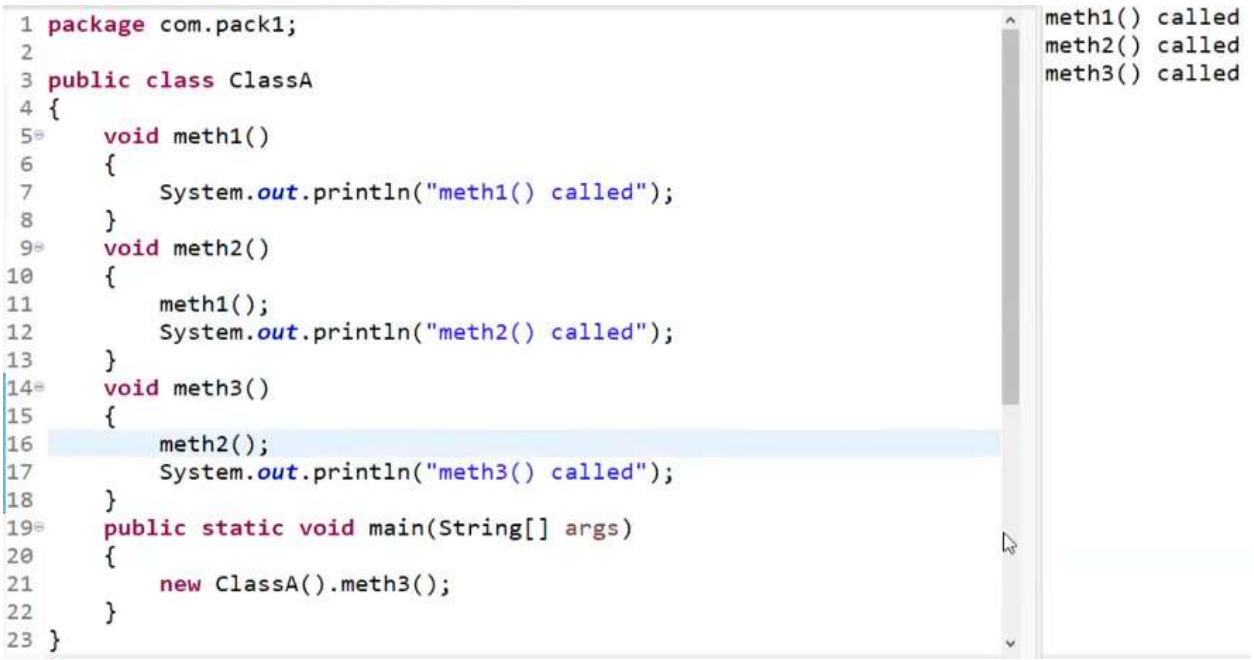
```java
public class ClassA
{
    int x=10;
    static int y=20;

    void meth1()
    {
        System.out.println("meth1() called");
        int x=111;
        int y=222;
        System.out.println("Instance Variable : "+this.x);
        System.out.println("Static Variable : "+ClassA.y);
        System.out.println("Local variable x: "+x+" y : "+y);
    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```

```
meth1() called
Instance Variable : 10
Static Variable : 20
Local variable x: 111 y : 222
```


Core Java Class-44
"this" key word

⇨ It is used to call present class methods.





Here we are writing meth1();

But internally the compiler consider this.meth1();

We cannot use ‘this’ key word in static areas, but by using ‘this’ key word we can call static method.

```
1  package com.pack1;
2
3  public class ClassA
4  {
5      void meth1()
6      {
7          System.out.println("meth1() called");
8      }
9      static void meth2()
10     {
11         this.meth1();
12         System.out.println("meth2() called");
13     }
14     void meth3()
15     {
16         meth2();
17         System.out.println("meth3() called");
18     }
19     public static void main(String[] args)
20     {
21         new ClassA().meth3();
22         //this.meth3(); // C.E because we cant use 'this' keyword inside a static area
23     }
24 }
```

```
1  package com.pack1;
2
3  public class ClassA
4  {
5      void meth1()
6      {
7          System.out.println("meth1() called");
8      }
9      static void meth2()
10     {
11         //this.meth1(); // C.E
12         System.out.println("meth2() called");
13     }
14     void meth3()
15     {
16         this.meth2();   warning line because the best way to call satic method is by using its class name
17         System.out.println("meth3() called");
18     }
19     public static void main(String[] args)
20     {
21         new ClassA().meth3();
22         //this.meth3(); // C.E because we cant use 'this' keyword inside a static area
23     }
24 }
```

```java
public class ClassA
{
    static void meth1()
    {
        System.out.println("meth1() called");
        //this.msg();// C.E
    }
    void meth2()
    {
        this.meth1();// We can call a static method with the help of this keyword but it is not recommended
        msg();// Compiler will by default add 'this' keyword
        System.out.println("meth2() called");
    }
    void meth3()
    {
        this.meth2();
        System.out.println("meth3() called");
    }
    void msg()
    {
        System.out.println("Java is awesome!!!");
    }
    public static void main(String[] args)
    {
        new ClassA().meth3();
        //this.meth3(); // C.E because we cant use 'this' keyword inside a static area
    }
}
```

```java
{
    static void meth1()
    {
        System.out.println("meth1() called");
        //this.msg();// C.E
    }
    void meth2()
    {
        this.meth1();// We can call a static method with the help of this keyword
        msg();// Compiler will by default add 'this' keyword
        System.out.println("meth2() called");
    }
    void meth3()
    {
        this.meth2();
        System.out.println("meth3() called");
    }
    void msg()
    {
        System.out.println("Java is awesome!!!");
    }
    public static void main(String[] args)
    {
        new ClassA().meth3();
        //this.meth3(); // C.E because we cant use 'this' keyword inside a static
    }
```

```
meth1() called
Java is awesome!!!
meth2() called
meth3() called
```

It is used to return the Instance of the present class

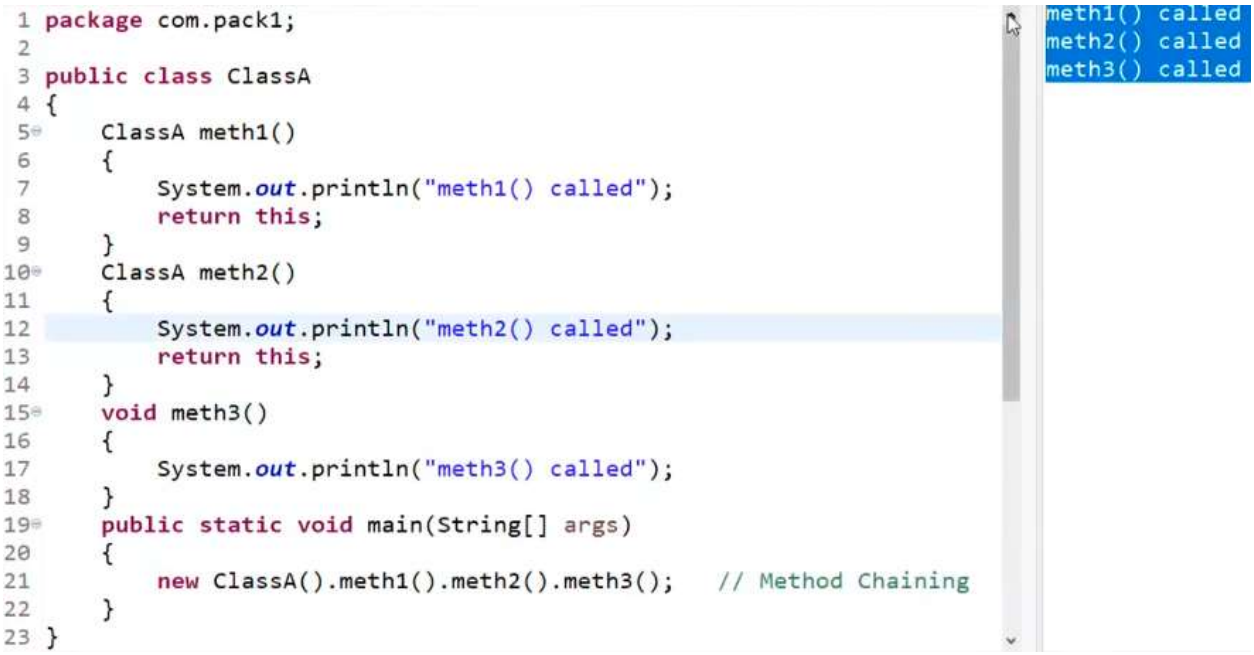
Here Access modifier is class it needs to return instance/object of present class

```java
package com.pack1;

public class ClassA
{
    ClassA meth1()
    {
        System.out.println("meth1() called");
        return new ClassA();
    }
    void meth2()
    {
        System.out.println("meth2() called");
    }
    void meth3()
    {
        System.out.println("meth3() called");
    }
    public static void main(String[] args)
    {
        new ClassA().meth1();
    }
}
```

```
meth1() called
```

```java
package com.pack1;

public class ClassA
{
    ClassA meth1()
    {
        System.out.println("meth1() called");
        return new ClassA();
    }
    ClassA meth2()
    {
        System.out.println("meth2() called");
        return new ClassA();
    }
    void meth3()
    {
        System.out.println("meth3() called");
    }
    public static void main(String[] args)
    {
        new ClassA().meth1().meth2().meth3();   // Method Chaining
    }
}
```

```
meth1() called
meth2() called
meth3() called
```



```
1  package com.pack1;
2
3  public class ClassA
4  {
5      ClassA meth1()
6      {
7          System.out.println("meth1() called");
8          return this;
9      }
10     ClassA meth2()
11     {
12         System.out.println("meth2() called");
13         return this;
14     }
15     void meth3()
16     {
17         System.out.println("meth3() called");
18     }
19     public static void main(String[] args)
20     {
21         new ClassA().meth1().meth2().meth3();    // Method Chaining
22     }
23 }
```

```
meth1() called
meth2() called
meth3() called
```

Instance and object both are same

Here new ClassA().meth1() means calling meth1 and but meth1 is returning instance/object

new ClassA().meth1().meth2() will becomes new ClassA().meth2() becomes meth1 is returning object

this	this()
1) It is a Keyword	1) It is a Constructor call
2) It is used to call present class variables & Methods	2) It is used to call present class Constructors.
3) We cant use 'this' keyword inside a static area.	3) We need to use this() constructor call **ONLY** inside a constructor, that too as a **FIRST** statement. If we are using this() anywhere else we will be getting **C.E**

```java
package com.pack1;

public class ClassA
{
    void meth1()
    {
        System.out.println("meh1() called");
    }
    ClassA()
    {
        System.out.println("Default constructor");
        new ClassA(10);
    }
    ClassA(int a)
    {
        System.out.println("Parameterized constructor : "+a);
    }
    public static void main(String[] args)
    {
        new ClassA();
    }
}
```

```
Default constructor
Parameterized constructor : 10
```

```java
package com.pack1;

public class ClassA
{
    void meth1()
    {
        System.out.println("meh1() called");
    }
    ClassA()
    {
        System.out.println("Default constructor");
        //new ClassA(10);
        this(10);
    }
    ClassA(int a)
    {
        System.out.println("Parameterized constructor : "+a);
    }
    public static void main(String[] args)
    {
        new ClassA();
    }
}
```

```java
package com.pack1;

public class ClassA
{
    void meth1()
    {
        System.out.println("meh1() called");
    }
    ClassA()
    {
        this(10);
        System.out.println("Default constructor");
        //new ClassA(10);

    }
    ClassA(int a)
    {
        System.out.println("Parameterized constructor : "+a);
    }
    public static void main(String[] args)
    {
        new ClassA();
    }
```

```
Parameterized constructor : 10
Default constructor
```

```java
public class ClassA
{
    void meth1()
    {
        System.out.println("meh1() called");
    }
    ClassA()
    {
        this(10);
        this.meth1();
        System.out.println("Default constructor");
        //new ClassA(10);
    }
    ClassA(int a)
    {
        System.out.println("Parameterized constructor : "+a);
    }
    public static void main(String[] args)
    {
        new ClassA();
    }
}
```

```
Parameterized constructor : 10
meh1() called
Default constructor
```

Assignment

```java
public class ClassA
{
        public int meth1(int a, int b)
        {
                System.out.println("meth1 called");
                System.out.println(b);
                System.out.println(100);
                return (a+b)+10+(a-b);
        }
        public int meth2()
        {
                System.out.println("meth2 called");
                return this.meth3()+8;
        }
        public ClassA()
        {
                this("Java is awesome");
                System.out.println(50);
        }
        int meth3()
        {
                System.out.println("meth3 called");
                return 10;
        }
```

```
public String meth4(int a, String s)
{
        System.out.println("meth4 called");
        a=a+4;
        System.out.println("a value===>"+a);
        System.out.println("s value===>"+s);
        return s+" is Object oriented programming language";
}
public int meth5(int a)
{
        System.out.println("meth5() called");
        return 10-a;
}
public ClassA(String s)
{
        String result=this.meth4(this.meth1(this.meth2(), this.meth5(5)+'A'-('a')), "Java");
        System.out.println(result);
        System.out.println(s);
}


        public static void main(String[] args)
        {
                new ClassA();
        }
}
```