DAY-32

28 July 2023 15:38

this KEYWORD

WHAT IS this KEYWORD

- IT IS A KEYWORD USED TO REFER THE CURRENT OBEJCT
- this KEYOWRD IS USED IN CONSTRUCTOR , NON STATIC BLOCK , NON STATIC METHODS BUT CAN NOT BE USED IN STATIC BLOCK OR STATIC METHODS

WHY IS this KEYWORD

- WHEN THE NON STATIC DATA MEMBER AND LOCAL VARIABLES HAVING SAME NAME/IDENTIFIERS. IN ORDER TO REFER /POINT TO NON STATIC VARIABLE **this** KEYWORD IS USED.

```
class Test1
{

        int x;
        public void m1()
        {
                int x = 5;
                System.out.println(x);
                System.out.println(this.x);

        }

        public static void main(String[] args)
        {
                Test1 a1 = new Test1();
                a1.m1();
        }
}
```

USING CONSTRUCTORS

```
class Test1
{

        int x;
        public Test1()
        {
                int x = 23;
                System.out.println(x);
                System.out.println(this.x);

        }

        public static void main(String[] args)
```

```
          {
                    Test1 a1 = new Test1();

          }
}
```

INTIALIAZING NON STATIC MEMBERS USING this KEYWORD

```
class Test1
{

          int x;
          public Test1(int x)
          {

                    this.x = x;
          }

          public static void main(String[] args)
          {
                    Test1 a1 = new Test1(10);
                    System.out.println(a1.x);

          }
}
```

MULTIPLE USE OF "this" KEYWORD

```
class Test1
{

          int x  =6;
          String s = "BitsQ";

          {
                    int x = 24;
                    System.out.println(x);
                    System.out.println("The non static value x is "+this.x);
                    System.out.println("The non static value s is "+this.s);
          }

          public static void main(String[] args)
          {
                    Test1 a1 = new Test1();


          }
}
```

SUPER KEYWORD

- IT IS USED TO REFER THE SUPER CLASS MEMBERS FROM THE CHILD CLASS , WHEN THE PARENT AND CHILD CLASS

MEMBERS ARE HAVING SAME NAME.

```
class Test1
{

	int x =5;

	public static void main(String[] args)
	{
		System.out.println("Good Evening");
	}


}

class Test2 extends Test1
{
	int x = 5;

	public void m1()
	{
		int x = 20;
		System.out.println(x);
		System.out.println("the x variable in parent class is "+ super.x);
	}
	public static void main(String[] args)
	{

		Test2 a1 = new Test2();
		a1.m1();

	}
}
```

NOTE :

- **this** KEYWORD AND **super** KEYWORD CAN BE USED IN THE SAME METHOD
- WE CAN USE **super , this** KEYWORD BOTH IN THE NON STATIC BLOCK, CONSTRUCTOR AND METHODS

CONSTRCUTOR CALLING

- THE PROCESS OF CALLING ONE CONSTRCUTOR FROM ANOTHER CONSTRCUTOR WITHIN SAME CLASS IS CALLED CONSTRUCTOR CALLING
- WE CAN PERFORM CONSTRUCTOR CALLING WITH THE HELP OF this().
- **this()** IS SPECIFICALLY USED FOR CONSTRUCTOR CALLING AND IT SHOULD BE FIRST STATEMENT OF CONSTRUCTOR BODY

```
class Test1
{

	public Test1()
	{
        System.out.println("inside the 1st Constructor");
        this(10);
    }

    public Test1(int a)
    {

        System.out.println("inside the 2nd Constructor");
    }



    public static void main(String[] args)
    {
        Test1 a1 = new Test1();
    }


}
```

- this() SHOULD ALWAYS BE THE FIRST STATEMENT OF CONSTRCUTOR BODY IN CASE OF CONSTRUCTOR CALLING.
- WE CAN NOT CALL MULTIPLE CONSTRUCTORS IN A SINGLE CONSTRUCTOR BODY.

ADVANTAGES

IT HELPS US TO EXECUTE ALL CONSTRUCTORS OF A CLASS BY CREATING A SINGLE OBJECT

CALLING MULTIPLE CONSTRCTORS

```
class Test1
{

    public Test1()
    {
        this(20);
        System.out.println("inside the 1st Constructor");

    }

    public Test1(int a)
    {
        this(23.4f, "BitsQ");
        System.out.println(a);
        System.out.println("inside the 2nd Constructor");
    }

    public Test1(float b, String c)
    {
        System.out.println(b);
        System.out.println(c);
        System.out.println("inside the 3rd Constructor");
    }
```

```
        public static void main(String[] args)
        {
                Test1 a1 = new Test1();
        }


}
```

NOTE :

This() and this KEYWORD CAN BE USED IN THE SAME CONSTRUCTOR

*******DIFFERENCE BETWEEN CONSTRUCTOR CALLING AND METHOD CALLING**

CONSTRUCTOR CALLING	**METHOD CALLING**
• CALLING ONE CONSTRUCTOR INSIDE ANOTHER CONSTRUCTOR BODY IS CALLED CONSTRCUTOR CALLING	• CALLING METHOD INSIDE ANOTHER METHOD BODY IS CALLED AS METHOD CALLING
• WE ACHIEVE IT USING this()	• WE CALL METHODS USING METHOD NAMES
• AT-LEAST SINGLE OBEJCT HAS TO BE CREATED	• AT-LEAST ONE OBEJCT HAS TO BE CREATED IN CASE OF NON STATIC METHODS
• CONSTRUCTOR CALLING STATEMENT HAS TO BE THE FIRST STATEMENT OF CONSTRUCTOR BODY.	• METHOD CALLING STATEMENT CAN BE WRITTEN ANYWHERE INSIDE METHOD BODY

******DIFFERENCE BETWEEN THIS AND THIS()**

this	this()
• this IS A KEYWORD USED TO REFFER CURRENT OBEJCT MEMBERS	• this() IS A METHOD USED TO PERFORM CONSTRCUTOR CALLING
• this CAN BE USED NON STATIC METHODS AND CONSTRUCTORS.	• this() CAN BE USED ONLY INSIDE CONSTUCTORS
• this CAN BE USED FOR MULTIPLE TIMES INSIDE A SAME CONSTRUCTOR /NON STATIC METHOD/NON STATIC BLOCK	• this() CAN'T BE USED MULTIPLE TIMES INSIDE THE SAME CONSTRCUTOR BODY
• this CAN BE USED ANYWHERE INSID CONSTRCUTOR /NON STATIC METHOD/NON STATIC BLOCK.	• this() HAS TO BE FIRST STATEMENT INSIDE THE CONSTRUCTOR BODY

- **TAKE AN EXAMPLE OF HOSPITAL MANAGEMENT SYSTEM.**

a. **FOR APPOINTMENT**
    i. NAME
    ii. AGE
    iii. NUMBER
    iv. DEPARTMENT

   v. GENDER

b. THEN HAVE CONSULTATION WITH Dr. NAME (CONSTRUCTOR OVERLOADING, CONSTUCTOR CALLING, this)