=> this keyword :-

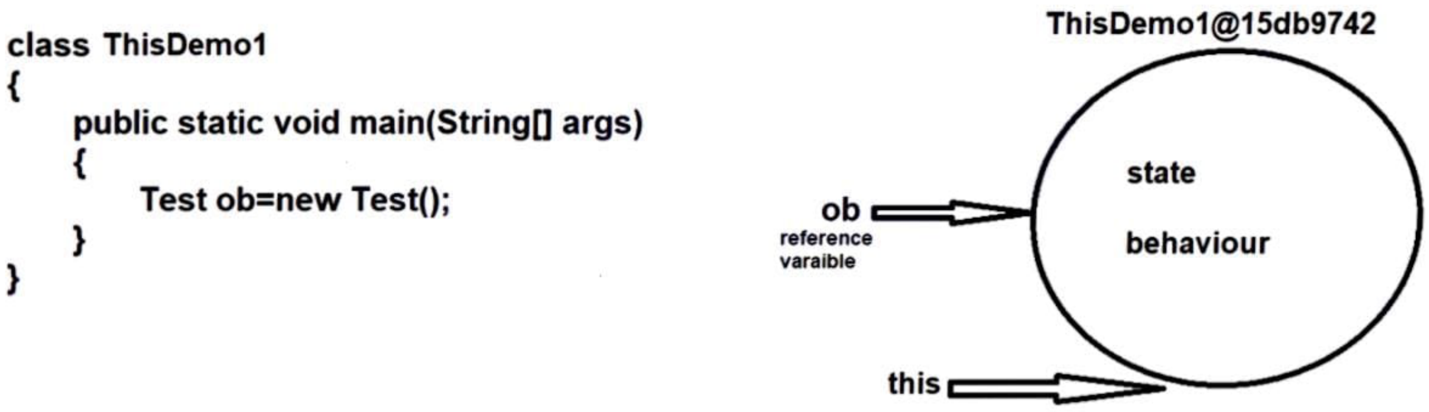
-> this keyword is "reference variable" that refers to the current object

Write a Program to prove that this keyword also refers to current object

```
public class ThisDemo1 {
    void m1()
    {
        System.out.println("I am in m1 method : "+this);
    }
    public static void main(String[] args)
    {
        ThisDemo1 ob=new ThisDemo1();
        System.out.println("I am in main method : "+ob);
        System.out.println(this);   // error(non-static variable this cannot be referenced from a static context)
        ob.m1();
    }
}
```

O/P:

```
I am in main method : StudentDemo.ThisDemo1@2a84aee7
I am in m1 method : StudentDemo.ThisDemo1@2a84aee7
```



```
class ThisDemo1
{
    public static void main(String[] args)
    {
        Test ob=new Test();
    }
}
```

```
class Student
{
   int rollno;
   String name;

   Student(int rollno, String name)
   {
      this.rollno=rollno;
      this.name=name;
   }

   void show()
   {
      System.out.println("Student Rollno : "+rollno);
      System.out.println("Student Name : "+name);
   }

   public static void main(String[] args)
   {
      Student s1=new Student(101, "deepak");
      s1.show();
   }
}
```

-> Use of this keyword :-

1. this keyword is used to refer the current class instance variable

-> this.instance_variable_name;

```
class ThisDemo2
{
    int no=10;

    void m1(int no)
    {
        System.out.println(no);
        System.out.println(this.no);
    }

    public static void main(String[] args)
    {
        ThisDemo2 ob=new ThisDemo2();
        ob.m1(20);
        System.out.println(ob.no);
    }
}
```

2. this keyword is used to invoke the current class method

-> this.methodName();

```
class ThisDemo4
{
    void m1()
    {
        System.out.println("i am in m1 method");
        m2();     // internally compiler is doing--> this.m2();
        //this.m2() ;
    }
    void m2()
    {
        System.out.println("i am in m2 method");
    }
    public static void main(String[] args)
    {
        ThisDemo4 ob=new ThisDemo4();
        ob.m1();
    }
}
```

3. this keyword is used to invoke the current class constructor

-> this();

-> this(-,-,-,-);

-> this keyword must be the first statement in the constructor call

```
class ThisDemo5
{
    ThisDemo5()
    {
        System.out.println("1");
    }
    ThisDemo5(int no)
    {
        this();
        System.out.println("2");
    }
    public static void main(String[] args)
    {
        ThisDemo5 ob2=new ThisDemo5(10);
         ThisDemo5 ob2=new ThisDemo5();
    }
}
=====================================================================
class ThisDemo6
{
    ThisDemo6()
    {
        this(10);
        System.out.println("1");
        //this(10); this is compile time error
    }
    ThisDemo6(int no)
    {
        System.out.println(no);
    }
    public static void main(String[] args)
    {
        ThisDemo6 ob2=new ThisDemo6();
    }
}
=======================================================================
```

constructor chaining:

```
class Test
{
    Test()
    {
        this(10);
        System.out.println("default constructor");
    }
    Test(int a)
    {
        this("deepak");
        System.out.println(a);
    }
    Test(String a)
    {
        System.out.println(a);
    }
}

class ConstructorChainingMain
{
    public static void main(String[] args)
```

4. this keyword can be used to pass as an argument in the method

```
class Test
{
    void m1()
    {
        System.out.println("1");
        // Test obj = new test() ;
        // m2(obj) ;
        m2(this);
    }
    void m2(Test t)
    {
        System.out.println(t);
    }
}

class ThisDemo7
{
    public static void main(String[] args)
    {
        Test t=new Test();
        t.m1();
        t.m2(this) ;    // error since this is a non static reference
    }
}
```

5. this keyword can be used to pass as an argument in the constructor

```
class Test
{
    void m1()
    {
        Xyz ob=new Xyz(this);
    }
}
class Xyz
{
    Xyz(Test t)
    {
        System.out.println(t);
    }
}

class ThisDemo8
{
    public static void main(String[] args)
    {
        Test t=new Test();
        t.m1();
    }
}
```

6. this keyword can be used to return current class instance

```
class Test
{
    Test m1()
    {
        return this;
    }
}
class ThisDemo9
{
    public static void main(String[] args)
    {
        Test ob=new Test();
        Test t=ob.m1();
        System.out.println(t);
    }
}
```