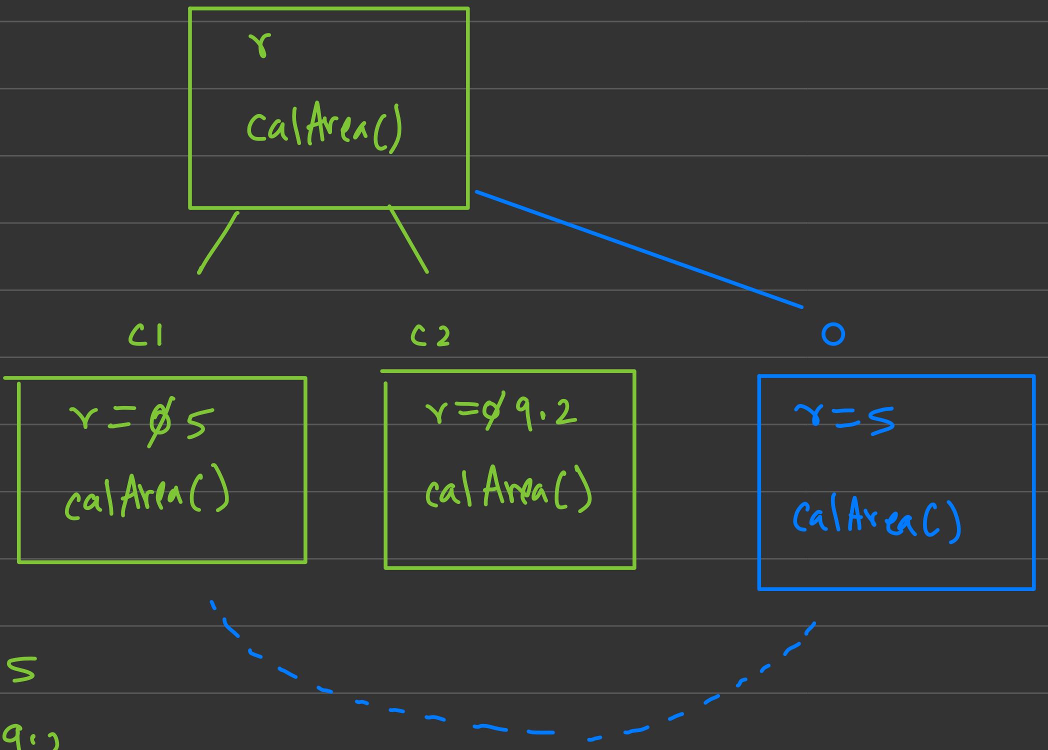
25 May 2023

Circle

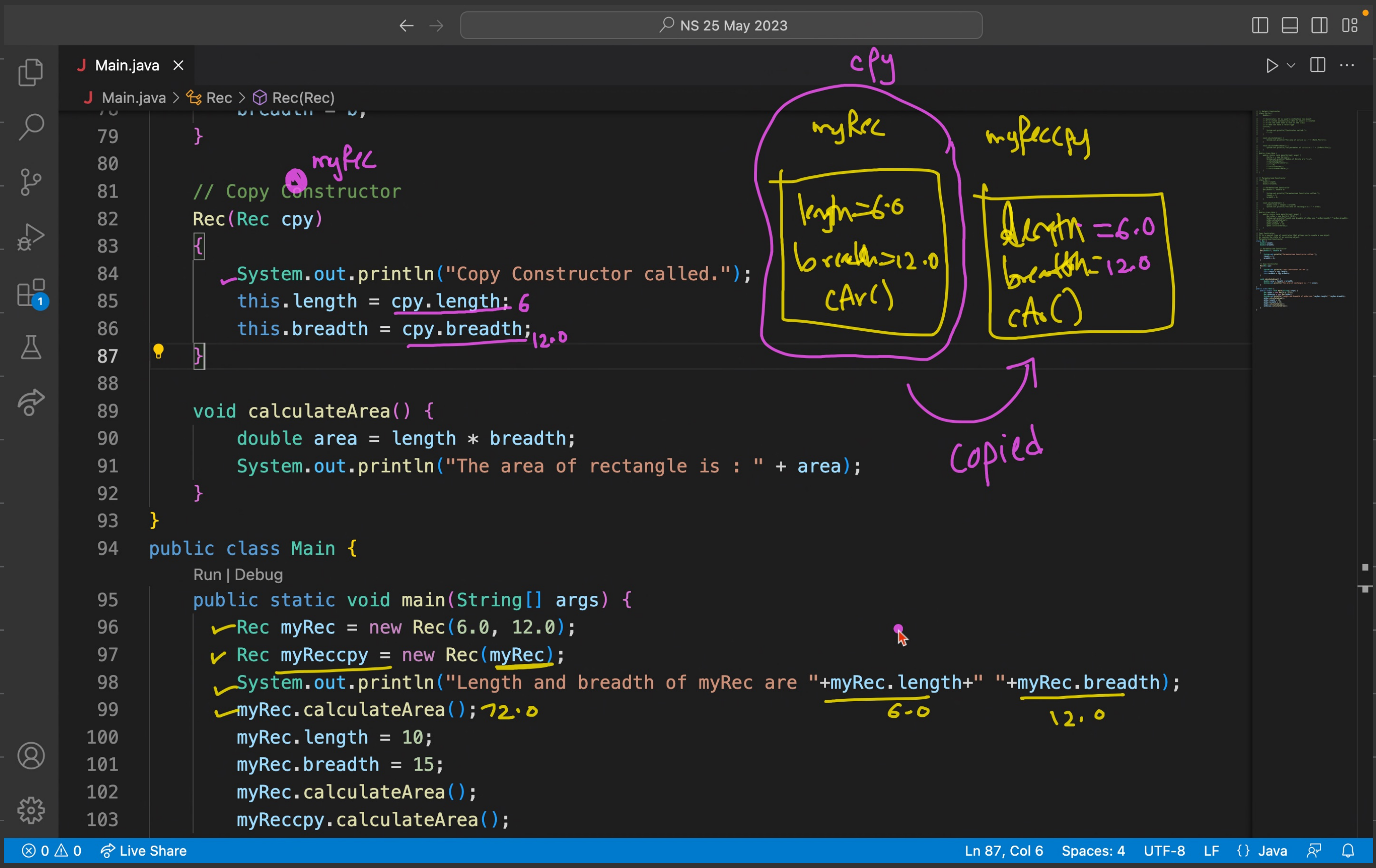


Create an object of circle class whose contents are same as c1.

c1.r=5

c2.r=9.2

c1.calArea() // 3.14X5X5



```
        breadth = b;
    }

    // Copy Constructor
    Rec(Rec cpy)
    {
        System.out.println("Copy Constructor called.");
        this.length = cpy.length;
        this.breadth = cpy.breadth;
    }

    void calculateArea() {
        double area = length * breadth;
        System.out.println("The area of rectangle is : " + area);
    }
}
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Rec myRec = new Rec(6.0, 12.0);
        Rec myReccpy = new Rec(myRec);
        System.out.println("Length and breadth of myRec are "+myRec.length+" "+myRec.breadth);
        myRec.calculateArea();
        myRec.length = 10;
        myRec.breadth = 15;
        myRec.calculateArea();
        myReccpy.calculateArea();
```

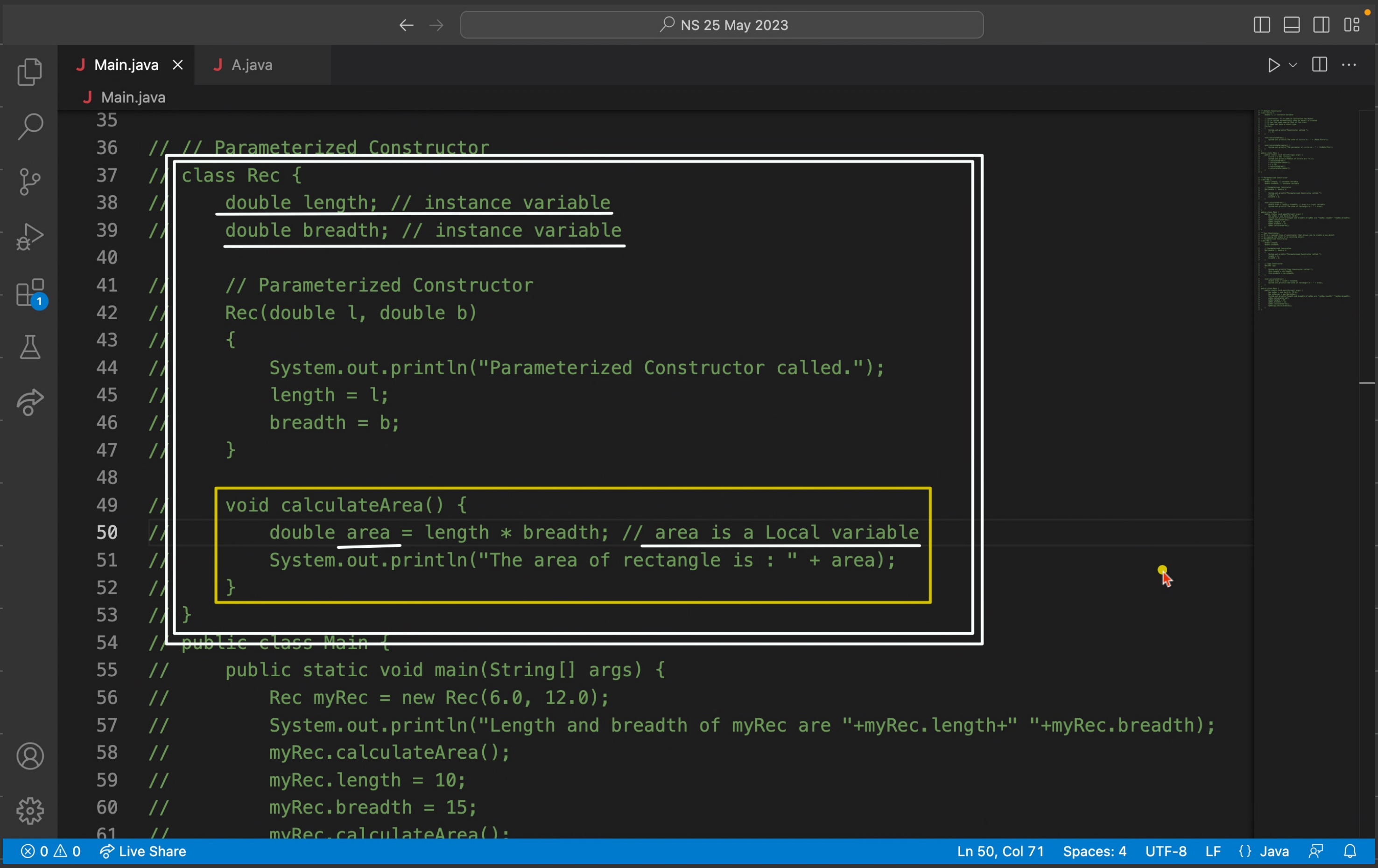
https://www.javatpoint.com/this-keyword

https://www.w3schools.com/java/ref_keyword_this.asp

Defn:

① **Instance Variable:** It is a variable which is declared inside a class but outside any of the methods.

② **Local Variable:**
- declared inside any of the mtd
- its value is available only inside the function where it is declared.



```java

// // Parameterized Constructor
// class Rec {
//     double length; // instance variable
//     double breadth; // instance variable

//     // Parameterized Constructor
//     Rec(double l, double b)
//     {
//         System.out.println("Parameterized Constructor called.");
//         length = l;
//         breadth = b;
//     }

//     void calculateArea() {
//         double area = length * breadth; // area is a Local variable
//         System.out.println("The area of rectangle is : " + area);
//     }
// }
// public class Main {
//     public static void main(String[] args) {
//         Rec myRec = new Rec(6.0, 12.0);
//         System.out.println("Length and breadth of myRec are "+myRec.length+" "+myRec.breadth);
//         myRec.calculateArea();
//         myRec.length = 10;
//         myRec.breadth = 15;
//         myRec.calculateArea();
```

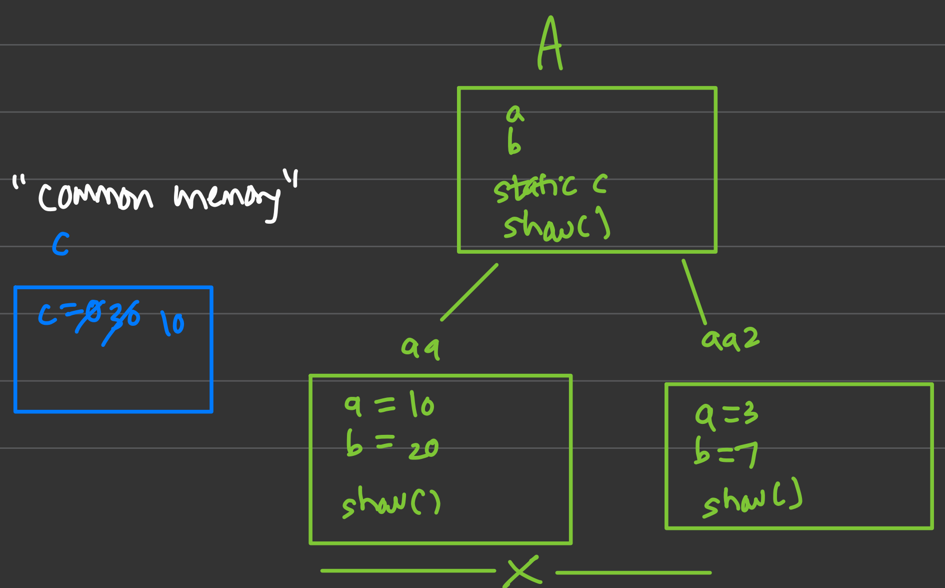
3 Static Variable:

1. When object of a class is created, it contains only list of non-static instance variables.

2. But if the instance variable is declared as static, only one copy of the static instance variable exists in the memory. And every object of the class shares the same copy of static instance variable.

3. To access the non-static instance variable and method of the class, we require the object of the same class.
Without object, these data are not accessible.

```
 class A
{ int a,b; //instance variables
  static int c; //static instance variable


  A(int x,int y)
  {
   a=x;
   b=y;
   c= a+b;
  }

  A()
  {a= 10;
   b=20;
   c=a+b;
  }

  void show()
  {Sopln("a="+a+" b="+b+"c="+c);
  }


  psvm()
  {

    A aa=new A();
    aa.show(); // 10,20,30
    A aa2 =new A(3,7);
    aa2.show(); // 3,7,10
    aa.show(); //10,20,10
  }
 }
```


A
a
b
static c
show()
"common memory"
c
c=~~0~~ ~~30~~ 10
aa
a = 10
b = 20
show()
aa2
a=3
b=7
show()