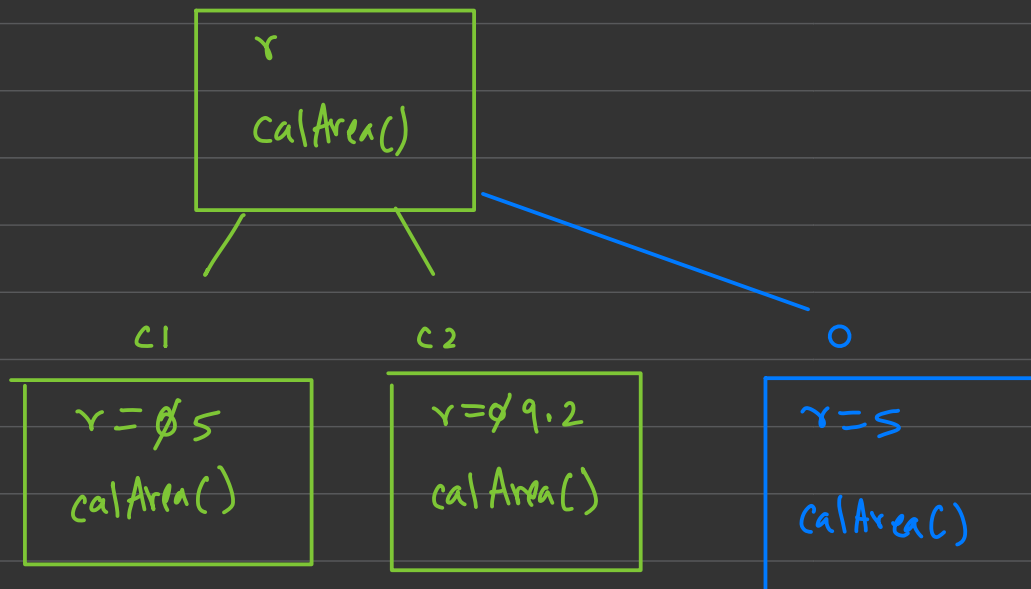


25 May 2023

Circle



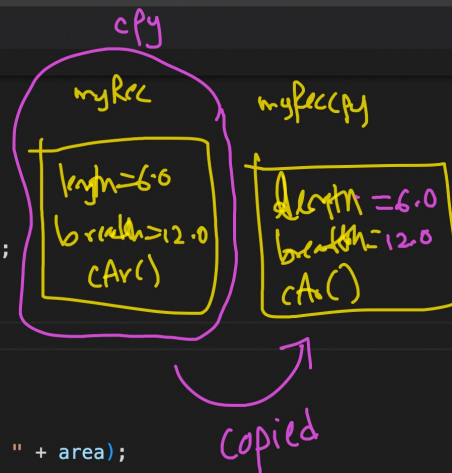
Create an object of circle class whose contents are same as  $c1$ .

$c1.r=5$   
 $c2.r=9.2$   
 $c1.calArea()$  //  $3.14 \times 5 \times 5$

```

Main.java x
Main.java > Rec > Rec(Rec)
78      breadth = 0;
79  }
80
81  // Copy Constructor
82  Rec(Rec cpy)
83  {
84      System.out.println("Copy Constructor called.");
85      this.length = cpy.length;
86      this.breadth = cpy.breadth;
87  }
88
89  void calculateArea() {
90      double area = length * breadth;
91      System.out.println("The area of rectangle is : " + area);
92  }
93  }
94  public class Main {
95      Run | Debug
96      public static void main(String[] args) {
97          ✓ Rec myRec = new Rec(6.0, 12.0);
98          ✓ Rec myRecCopy = new Rec(myRec);
99          ✓ System.out.println("Length and breadth of myRec are "+myRec.length+" "+myRec.breadth);
100         ✓ myRec.calculateArea();
101         myRec.length = 10;
102         myRec.breadth = 15;
103         myRec.calculateArea();
104         myRecCopy.calculateArea();

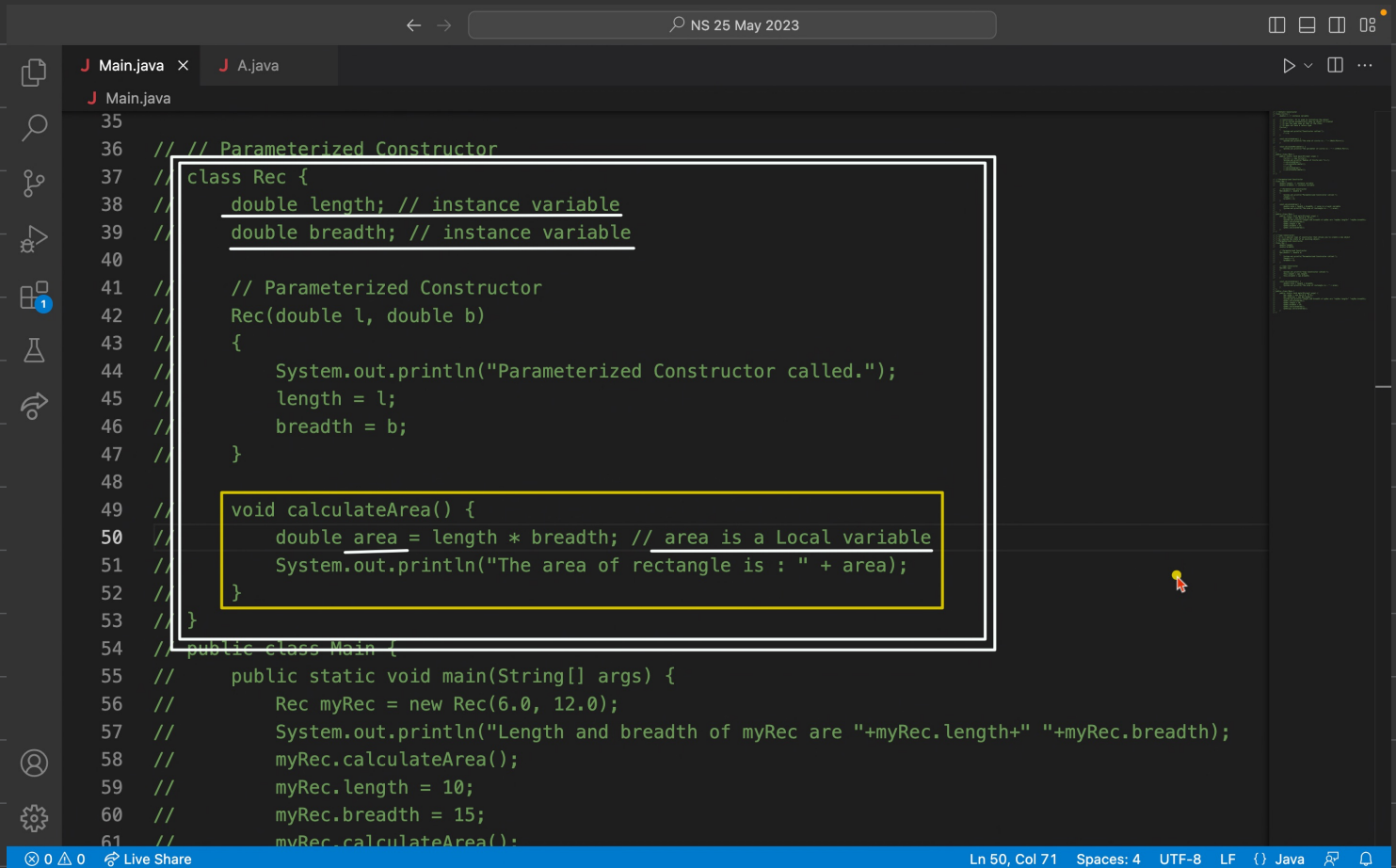
```



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.



The screenshot shows a Java IDE with two files: Main.java and A.java. The Main.java file contains the following code:

```
35
36 // // Parameterized Constructor
37 // class Rec {
38 //     double length; // instance variable
39 //     double breadth; // instance variable
40
41 //     // Parameterized Constructor
42 //     Rec(double l, double b)
43 //     {
44 //         System.out.println("Parameterized Constructor called.");
45 //         length = l;
46 //         breadth = b;
47 //     }
48
49 //     void calculateArea() {
50 //         double area = length * breadth; // area is a Local variable
51 //         System.out.println("The area of rectangle is : " + area);
52 //     }
53 // }
54 // public class Main {
55 //     public static void main(String[] args) {
56 //         Rec myRec = new Rec(6.0, 12.0);
57 //         System.out.println("Length and breadth of myRec are "+myRec.length+" "+myRec.breadth);
58 //         myRec.calculateArea();
59 //         myRec.length = 10;
60 //         myRec.breadth = 15;
61 //         myRec.calculateArea();
```

The code is annotated with two boxes:

- A white box highlights the class-level variables: `double length; // instance variable` and `double breadth; // instance variable`.
- A yellow box highlights the method-level variable: `double area = length * breadth; // area is a Local variable`.

The IDE interface includes a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Testing, and Settings. The bottom status bar shows "Ln 60, Col 71", "Spaces: 4", "UTF-8", "LF", and "Java".

### 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()
    {
        System.out.println("a=" + a + " b=" + b + " c=" + c);
    }

    public static void main()
    {
        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
    }
}
```

