

## $\Rightarrow$ class and object

instance var

local var

ref variable

JVM Data areas (Stack and Heap)

(JVM)

class Launch

```
{  
    public void main (String [ ] args)  
    {  
        System.out.println ("Hello");  
    }  
}
```

y

classes and objects:

class Student

```
{  
    String name ;  
    int age ;  
    int marks ;  
    void study ()  
}  
}
```

Student (st) = new Student () ;

st . study () ;

st . name = "Rohan" ;

ref =  
obj  
not

(student)

```

class Dog
{
    String nome;
    int cost;
    void eat();
    void sleep();
    d = new Dog();
}

```

Dog d = new Dog();  
d.eat();  
d.sleep();  
d → name  
cost

```

class Student
{
    int age;
    String name;
}

```

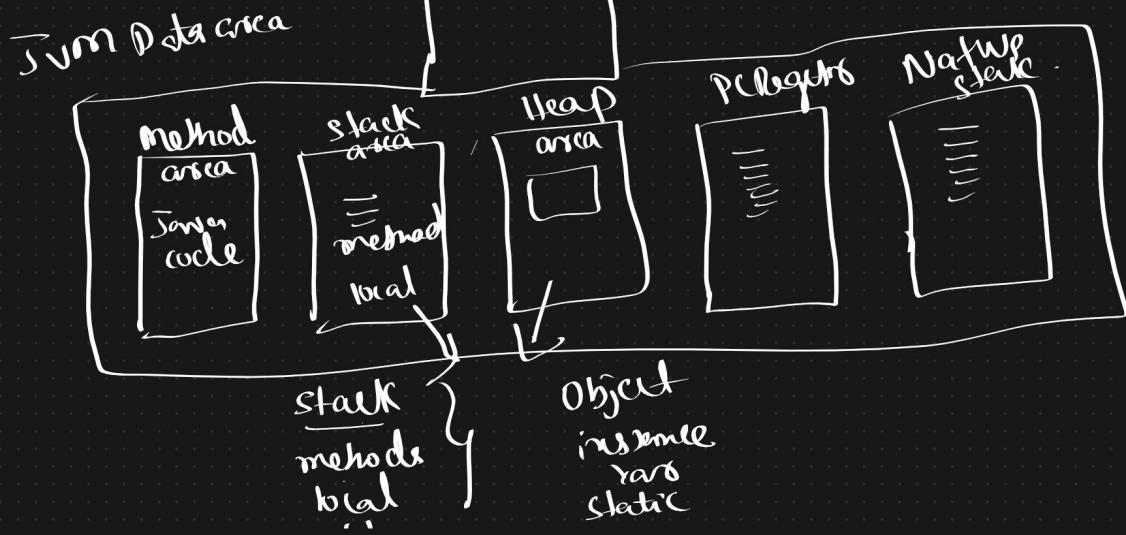
```

class Student
{
    void disp();
    d =
}

```

JVM ⇒ JRE

java → javac → .class → JVM ⇒  
Bytecodes



# Instance variable { Local variable

int age = 18;



local variable

instance  
var

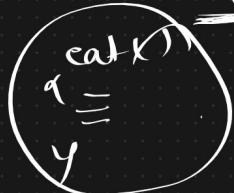
class Demo

```
{
    int age; // instance var
    int cost;
    String name;
    void dupl()
    {
        System.out.println();
    }
}
```

class Demo

```
{
    void dupl()
    {
        int age = 18; // local var
        int cost = 0;
        for (int i = 0; i < 10; i++)
    }
}
```

new Dog();



stack area



Heap area



class Dog

Dog d = new Dog();  
dog d = new Dog(); X  
Dog d = new Dog();

$\Rightarrow$  classes and object  $\Rightarrow$  ✓  
Heap & Stack  $\Rightarrow$  ✓       $\text{Dom}(\text{J}) = \text{new Dom}()$   
 $\left\{ \begin{array}{l} \text{instances}, \text{locals} \Rightarrow \checkmark \\ \text{ref var} \end{array} \right.$

## method (or) function

In Java  $\Rightarrow$  method  $\Rightarrow$  Task (or) activity  
methods



4  $\Rightarrow$  things

output name (input)

{

==== Activity | task

$\Rightarrow$  {

y

④ return type ① ②  
return type (parameter)

③  
==== Body

void add()

{

====

④ ① ②  
int add()

{

====

y

③  
return 10;

y

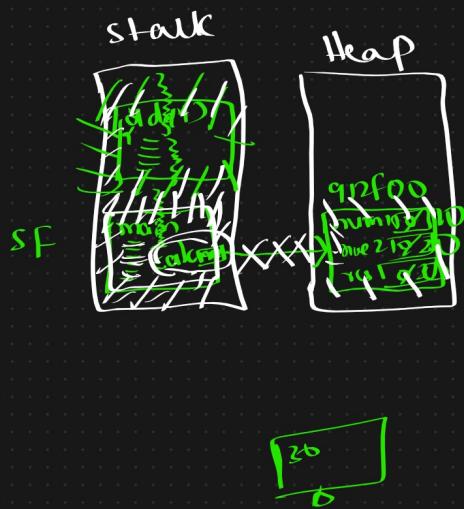
```

class Calculator
{
    int num1;
    int num2;
    int res;

    void add()
    {
        num1=10;
        num2=20;
        res=num1+num2;
        System.out.println(res);
    }
}

public class LaunchMethod
{
    public static void main(String[] args)
    {
        Calculator calc=new Calculator();
        calc.add();
    }
}

```



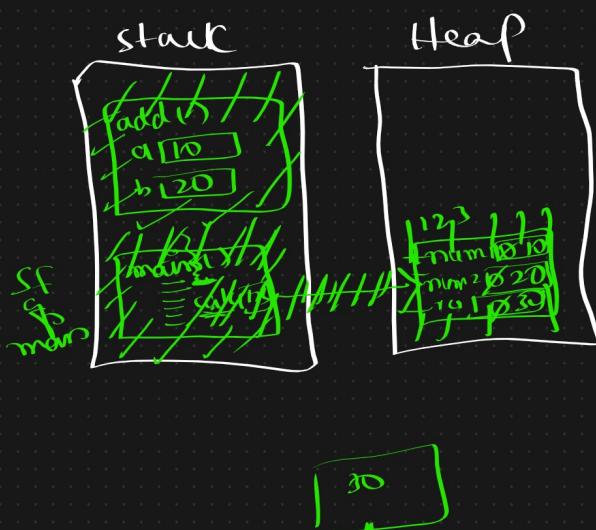
```

class Calculator
{
    int num1;
    int num2;
    int res;

    void add(int a, int b)
    {
        num1=a;
        num2=b;
        res=num1+num2;
        System.out.println(res);
    }
}

public class LaunchMethod
{
    public static void main(String[] args)
    {
        Calculator calc=new Calculator();
        //calc.add(); 
        calc.add(10,20);
    }
}

```



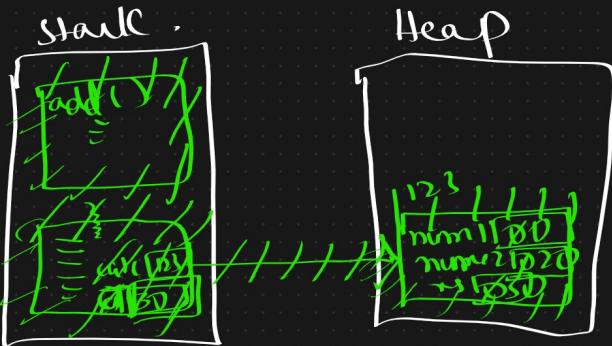
```

class Calculator
{
    int num1;
    int num2;
    int res;

    int add()
    {
        num1=10;
        num2=20;
        res=num1+num2;
        return res;
    }
}

public class LaunchMethod
{
    public static void main(String[] args)
    {
        Calculator calc=new Calculator();
        //calc.add();
        //calc.add(10,20);
        int a=calc.add();
        System.out.println(a);
    }
}

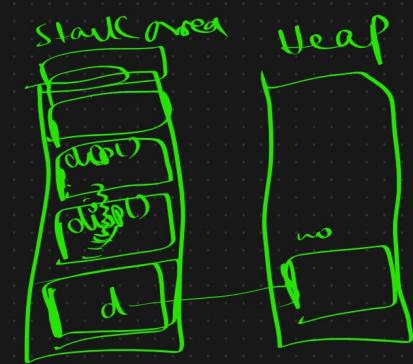
```



```

class Demo1
{
    void disp()
    {
        disp();
    }
}

```



```

public class LaunchSOF {
    public static void
    main(String[] args)
    {
        Demo1 d=new Demo1();
        d.disp();
    }
}

```