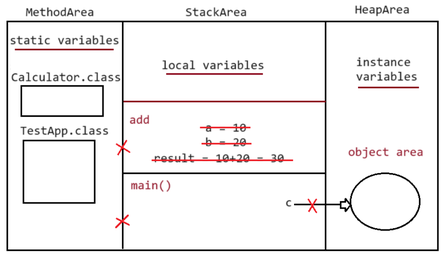
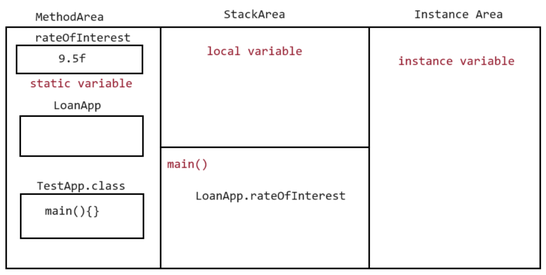


Local variables

=>variables declared inside the method
=>Memory will be given inside stack area
=>Once the control enters inside the method memory will be given.
=>Once the control leaves the method memory will be taken out.
=>No default value will be given to the local variables

```
class LoanApp
{
    //static variable
    static float rateOfInterest = 9.5f;
}
public class TestApp
{
    public static void main(String[] args)
    {
        System.out.println(LoanApp.rateOfInterest);
    }
}
```



static variable

a. Memory will be given in the Method-Area.
b. Memory will given at the time of loading .class file
c. Default value will be given if user won't specify any value.
d. Memory will be taken out at the time of unloading the .class file.
e. Static variables can be accessed in 2 ways
    a. using ClassName.
    b. using reference of the object.

```
class Student
{
    //instance variable
    String name;
    int age;

    //constructor
    Student(String name ,int age){
        this.name = name;
        this.age  = age;
    }

    //instance method
    public void disp(){
        System.out.println("Name is :: "+name);
        System.out.println("Age  is :: "+age);
    }
}

public class TestApp
{
    public static void main(String[] args)
    {
        Student std1 = new Student("sachin",51);
        std1.disp();

        Student std2 = new Student("kohli",37);
        std2.disp();
    }
}
```


Output

```
D:\Decode Java1.0Batch>javac TestApp.java
D:\Decode Java1.0Batch>java TestApp
Name is :: sachin
Age  is :: 51

Name is :: kohli
Age  is :: 37
```