



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore - 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

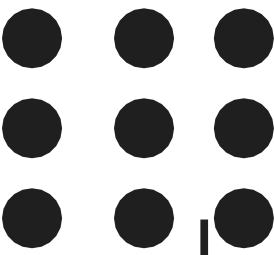
19CS204 OBJECT ORIENTED PROGRAMMING

I YEAR /II SEMESTER

Unit 1- INTRODUCTION TO OOP

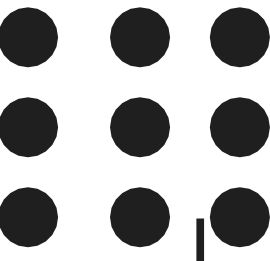
Static Members





Static Members

- There will be times when you will want to define a class member that will be used independently of any object of that class.
- Normally, a class member must be accessed only in conjunction with an object of its class.
- However, it is possible to create a member that can be used by itself, without reference to a specific instance.
- To create such a member, precede its declaration with the keyword **static**.

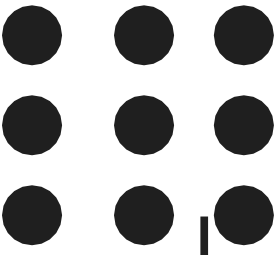


Static Members

- When a member is declared static, it can be accessed before any objects of its class are created, and without reference to any object.
- You can declare both methods and variables to be static.
- The most common example of a static member is `main()`.
- `main()` is declared as static because it must be called before any objects exist.



Static Members

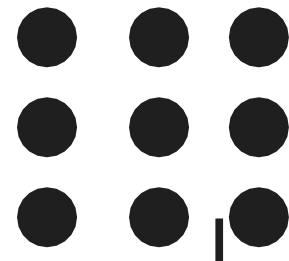


Methods declared as static have several restrictions:

- They can only directly call other static methods.
- They can only directly access static data.
- They cannot refer to this or super in any way.



Static Members



```
public class staticstudent
{
    static int age=25;
    static String name;
    static int mark1=40,mark2;
    static void display(String n)
    {
        name=n;
        System.out.println("Name is " +name);
    }
    static{
        mark2= mark1 * 2;
        System.out.println("Mark 2 is " +mark2);
    }
    public static void main(String[] args)
    {
        display("Ryan");
        System.out.println("Age is "+age);
    }
}
```



Static Members



Accessing outside of class

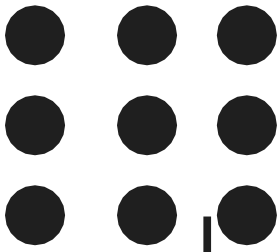
- Outside of the class in which they are defined, static methods and variables can be used independently of any object.
- To do so, you need only specify the name of their class followed by the dot operator.
- For example, if you wish to call a static method from outside its class, you can do so using the following general form:

classname.method()

- Here, classname is the name of the class in which the static method is declared.

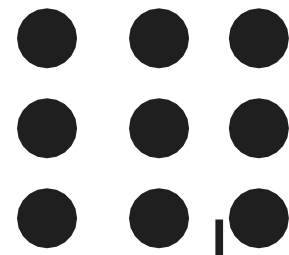


Static Members



```
class staticstudent
{
    static int age=40;
    static String name;
    static void display(String n)
    {

        name=n;
        System.out.println("Name is " +name);
    }
}
public class student
{
    public static void main(String[] args)
    {
        staticstudent.display("Ryan");
        System.out.println("Age is " +staticstudent.age);
    }
}
```



THANK YOU