



# **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**

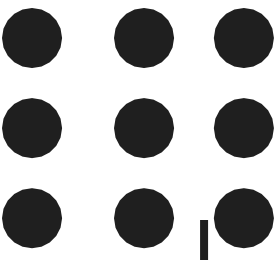
**Topic - Nested Interface**





# Nested Interface

- An interface which is declared inside another interface or class is called nested interface.
- They are also known as inner interface.
- Since nested interface cannot be accessed directly, the main purpose of using them is to resolve the namespace by grouping related interfaces (or related interface and class) together.
- This way, we can only call the nested interface by using outer class or outer interface name followed by dot( . ), followed by the interface name



# Nested Interface

- Nested interfaces are static by default. You don't have to mark them static explicitly as it would be redundant.
- Nested interfaces declared inside class can take any access modifier,
- Nested interface declared inside interface is public implicitly



# Nested Interface



**Nested interface declared inside another interface**

```
interface InterfaceA {
    void display();
    interface InterfaceB {
        void myMethod();
    }
}

public class NestedInterfaceDemo1 implements InterfaceA.InterfaceB {
    public void myMethod() {
        System.out.println("Nested interface method");
    }

    public static void main(String args[]) {
        InterfaceA.InterfaceB obj = new NestedInterfaceDemo1();
        obj.myMethod();
    }
}
```



# Nested Interface



**Nested interface declared inside a class**

```
class MyClass{
    interface InterfaceB {
        void myMethod();
    }
}

public class NestedInterfaceDemo2 implements MyClass.InterfaceB {
    public void myMethod() {
        System.out.println("Nested interface method");
    }

    public static void main(String args[]){
        MyClass.InterfaceB obj= new NestedInterfaceDemo2();
        obj.myMethod();
    }
}
```



**THANK YOU**