



Present by
Tofayel Ahamed Topu
.....
Id: CSE2202026024
Section: 27M1
Course Title: State of Art Programming Java

**Submitted to
Sahidulla Ayon (sir)**

Sonargaon University

INHERITANCE INTERFACE

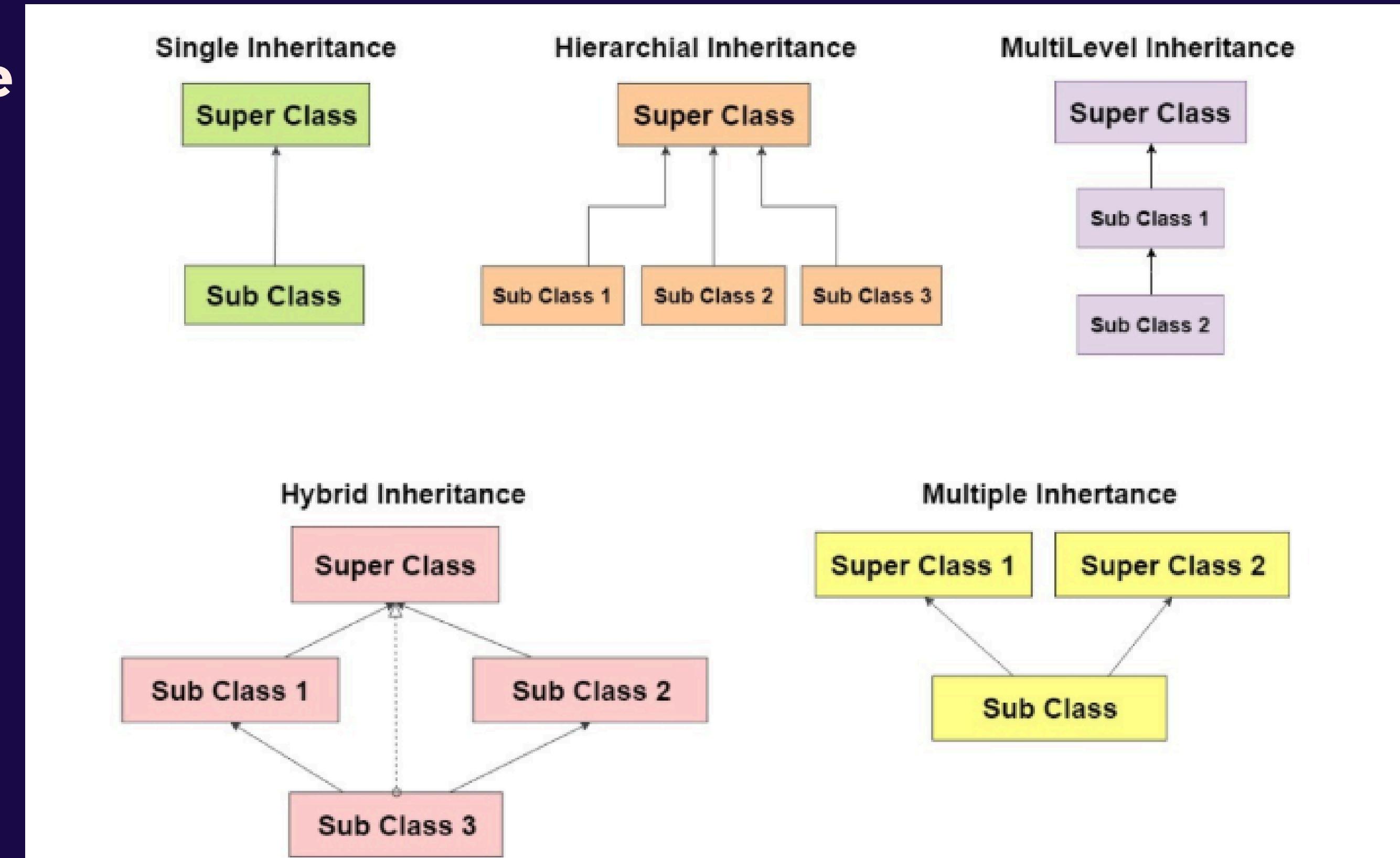
A futuristic space station interior is shown, featuring two astronauts in full space suits. One astronaut on the left is wearing a pink suit and is focused on reading a book. Another astronaut on the right is wearing a blue suit and is also reading a book. The background is dark with glowing purple structural elements and a central circular light source. The overall theme is a blend of education and technology in a space environment.

What is Inheritance

Inheritance is an important OOP Concept in Java, through which a class inherits the properties and behaviors (methods) of another class.

Type of Inheritance

- Single Inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Multilevel Inheritance
- Hybrid Inheritance



-> Single Inheritance

```
1 /*
2  Authour: Tofayel Ahmd Topu
3  ID:      : CSE2202026024
4 */
5
6 // Parent class
7 class Animal {
8     void eat() {
9         System.out.println("This animal eats food.");
10    }
11 }
12
13 // Child class
14 class Dog extends Animal {
15     void bark() {
16         System.out.println("Dog barks");
17    }
18 }
19
20 // Main class
21 public class Test {
22     public static void main(String[] args) {
23         Dog d = new Dog();
24         d.eat();
25         d.bark();
26     }
27 }
28 }
```

Benefits

Code Reusability - Code from previous classes can be reused

Maintenance Ease - No need to write the same code over
and over

Extensibility - New features can be added easily

Explanation

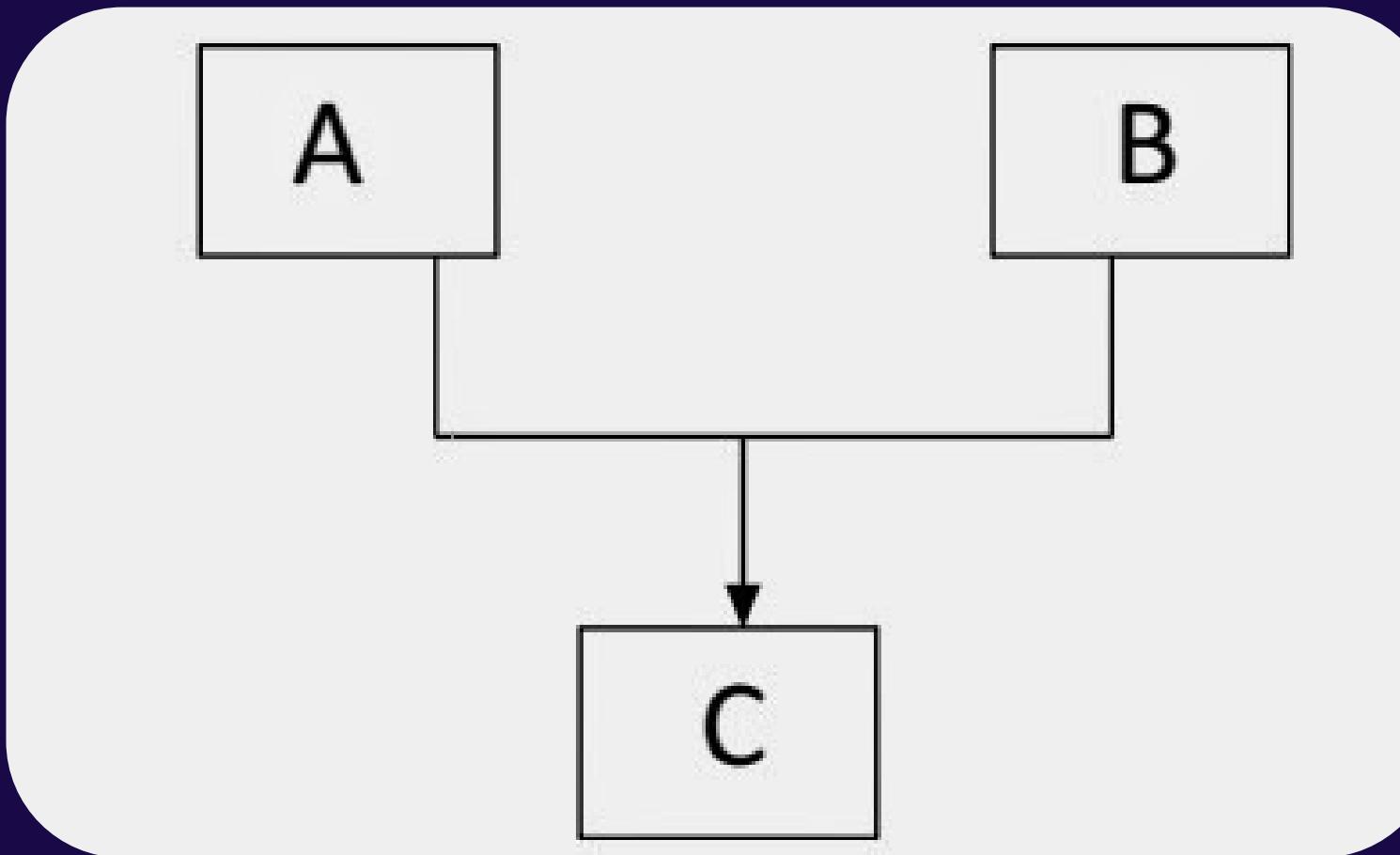
The Dog class inherits from Animal.

**The Dog class itself has created the bark()
method.**

**But the sound() method is inherited from
the Animal class.**

-> Multiple Inheritance

Output



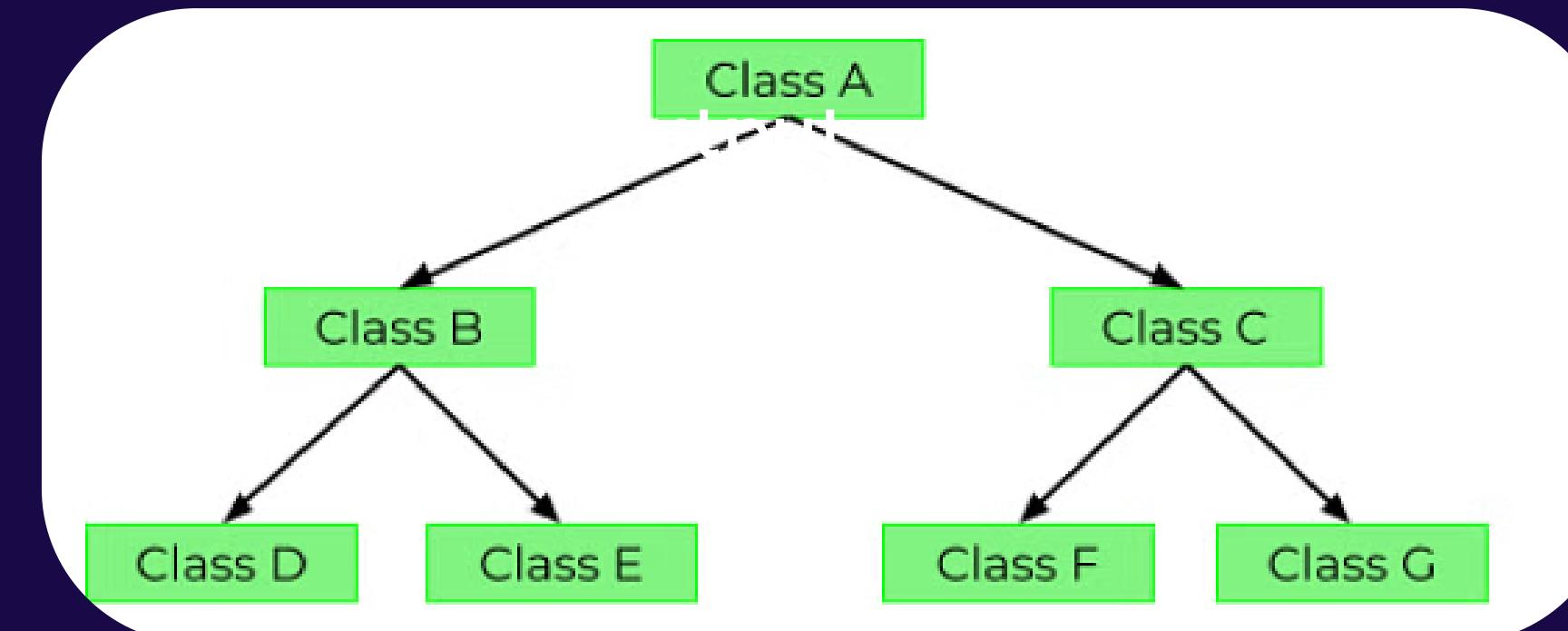
```
[topu@kali) - [~/De
$ cd /home/topu/De
ceptionMessages -cp /
This is A
This is B
```

```
1 /*
2  Authour: Tofayel Aham Topu
3  ID:      : CSE2202026024
4 */
5
6 interface A {
7     void show();
8 }
9
10 interface B {
11     void display();
12 }
13
14 class C implements A, B {
15     @Override
16     public void show() {
17         System.out.println("This is A");
18     }
19
20     @Override
21     public void display() {
22         System.out.println("This is B");
23     }
24 }
25 public class App {
26     public static void main(String[] args) {
27         C obj = new C();
28         obj.show();
29         obj.display();
30     }
31 }
```



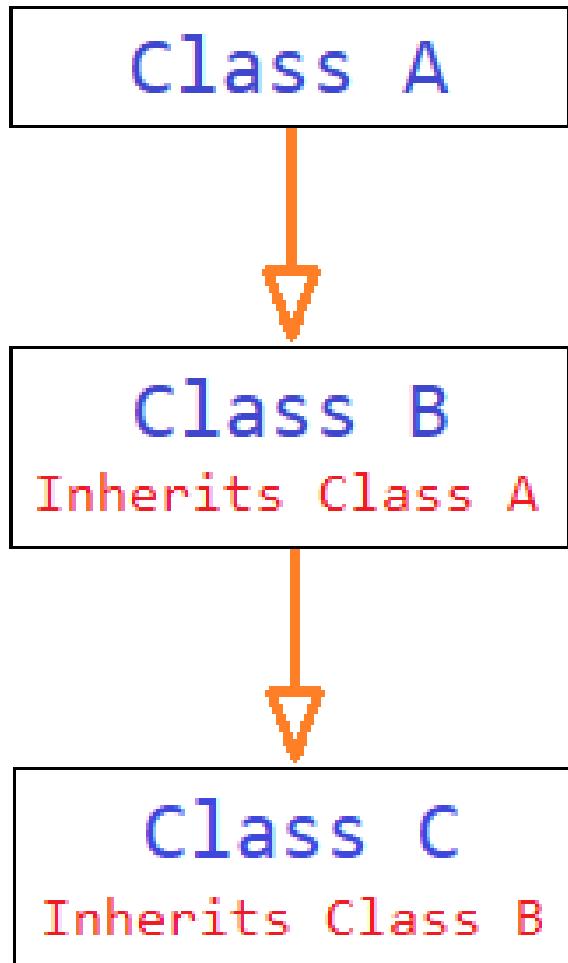
```
1  /*
2   Author: Tofayel Ahmd Topu
3   ID:    : CSE2202026024
4   */
5
6 class Animal {
7     void sound() {
8         System.out.println("Animals make sound");
9     }
10}
11
12 class Dog extends Animal {
13     void bark() {
14         System.out.println("Dog barks");
15     }
16}
17
18 class Cat extends Animal {
19     void meow() {
20         System.out.println("Cat meows");
21     }
22}
23 @SuppressWarnings("unused")
24 class Main {
25     public static void main(String[] args) {
26         Dog d = new Dog();
27         d.sound(); // Parent method
28         d.bark(); // Child method
29
30         Cat c = new Cat();
31         c.sound(); // Parent method
32         c.meow(); // Child method
33     }
34}
```

→ Hierarchical Inheritance



```
(topu㉿kali)-[~/Desktop/Multiple_Inheritance]
└─$ /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin
    java Main
    Animals make sound
    Dog barks
    Animals make sound
    Cat meows
```

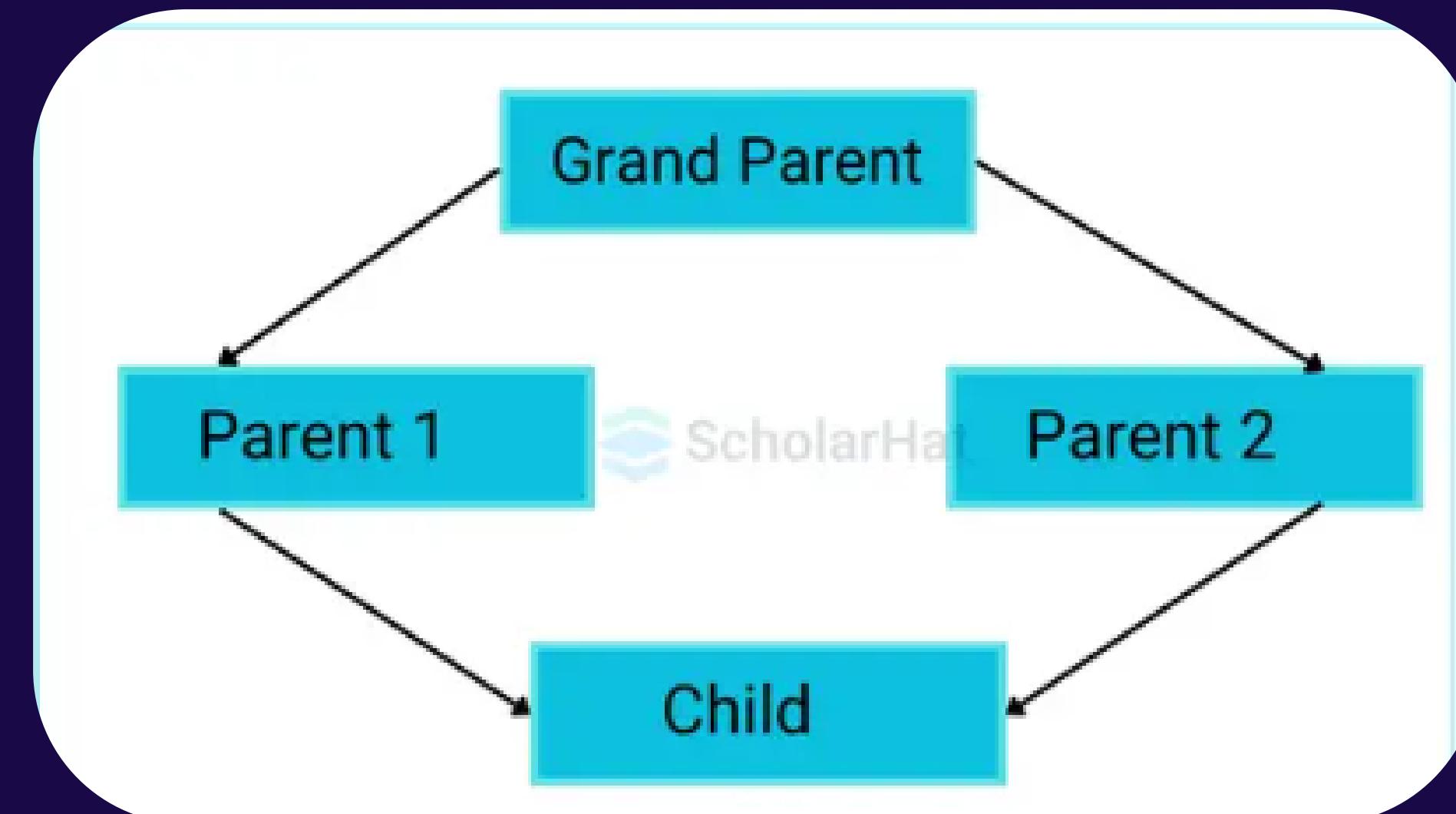
-> Multilevel Inheritance



```
—(topu㉿kali)-[~/Desktop/Multi  
└$ cd /home/topu/Desktop/Multi  
ultiple_Inheritance/bin Test  
Animal eats food  
Dog barks  
Puppy weeps
```

```
1  /*  
2  Authour: Tofayel Ahmd Topu  
3  ID:      : CSE2202026024  
4  */  
5  
6  class Animal {  
7      void eat() {  
8          System.out.println("Animal eats food");  
9      }  
10 }  
11  
12 class Dog extends Animal {  
13     void bark() {  
14         System.out.println("Dog barks");  
15     }  
16 }  
17  
18 class Puppy extends Dog {  
19     void weep() {  
20         System.out.println("Puppy weeps");  
21     }  
22 }  
23  
24 public class Test {  
25     public static void main(String[] args) {  
26         Puppy p = new Puppy();  
27         p.eat();    // Grandparent class method  
28         p.bark();   // Parent class method  
29         p.weep();   // Own method  
30     }  
31 }  
32 }
```

-> Hybrid Inheritance



```
1 /*  
2  Author: Tofayel Ahamed Topu  
3  ID:      : CSE2202026024  
4 */  
5  
6 class Animal {  
7  
8     // parent class  
9     String name;  
10    public void eat() {  
11        System.out.println("I can eat");  
12    }  
13 }  
14  
15 // inherit from Animal  
16 class Dog extends Animal {  
17  
18     // new method in subclass  
19     public void display() {  
20         System.out.println("My name is " + name);  
21     }  
22 }  
23  
24 @SuppressWarnings("unused")  
25 class MainExample {  
26     public static void main(String[] args) {  
27  
28         Dog labrador = new Dog();  
29  
30         labrador.name = "Topu";  
31         labrador.display();  
32         labrador.eat();  
33     }  
34 }
```

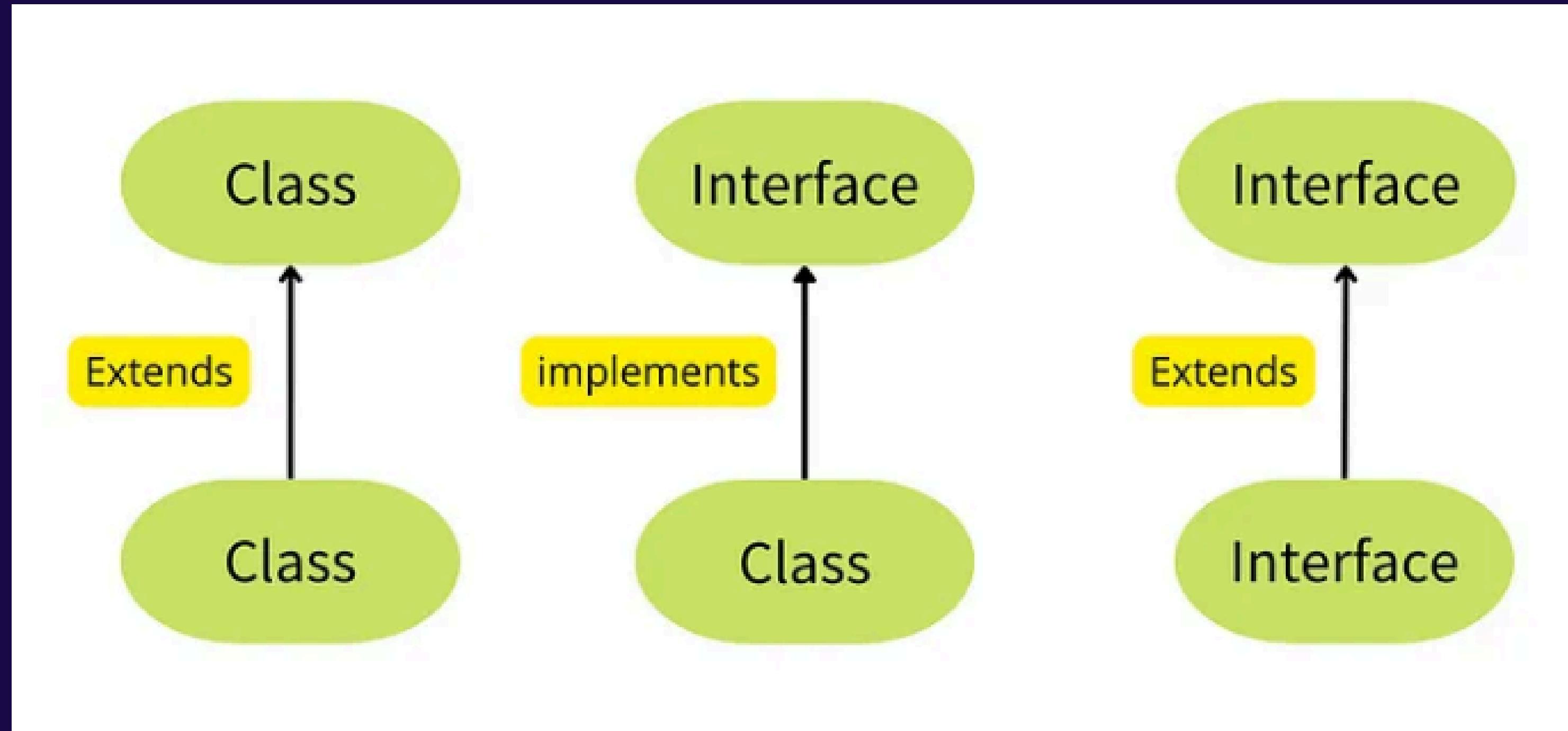
```
(topu㉿kali)-[~/Desktop/Multi]  
● └─$ /usr/bin/env /usr/lib/jvm/ja
```

```
My name is Topu  
I can eat
```



What is an Interface?

In Java, an Interface is a completely abstract class that contains only the method signature (it means, method name, return type, parameters) and no body or code.



```
1  /*
2  Authour: Tofayel Aham Topu
3      ID: CSE2202026024
4  */
5
6 // create an interface
7 interface Campus {
8
9     void getName(String name);
10}
11
12// class implements interface
13class Sonargaon_university implements Campus {
14
15    // implementation of abstract method
16    public void getName(String name) {
17        System.out.println("Sonargaon university: " + name);
18    }
19}
20
21class Main {
22
23    public static void main(String[] args) {
24        Sonargaon_university Campus = new Sonargaon_university();
25        Campus.getName("Student");
26    }
27}
28
```

Output

```
(topu@kali)-[/media/.../All_Program/SU/9th_semester/Java Programming]
$ Sonargaon university: Student
∅ ▷ ⌂ ⌂ Indexing completed.
```

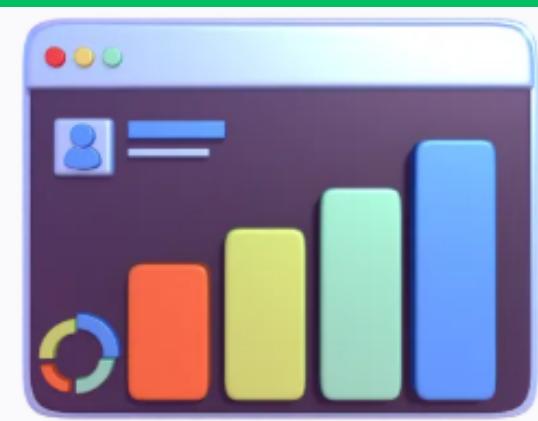
Features of the Interface

Contains only method declarations

All methods are public and abstract

All variables are public, static, and final

Supports multiple inheritance



Why are interfaces used?

To achieve polymorphism

To take advantage of multiple inheritance

To increase loose coupling and scalability





topum800@gmail.com



+88 01310260638

THANK YOU!