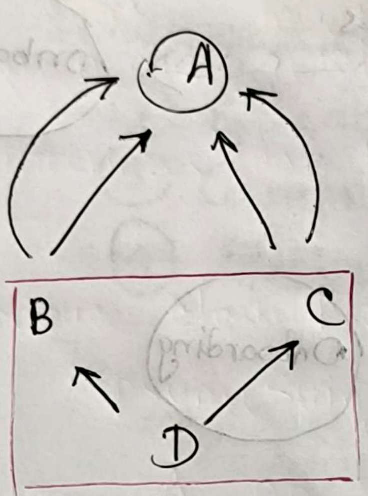
Advanced Programming

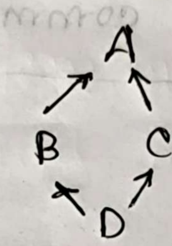
CSE 342 | 29-01-2026 | Class no. 08

Multi-level Inheritance

Multiple Inheritance → using interface



This will cause error (normally)



"Diamond Problem"

↳ ambiguity problem

↳ solved using virtual class

Solving Diamond Problem using Interface

Interface → abstract classএর মতো

- → "Complete Abstract Class"
- → before JDK7
  - ↳ only abstract methods can be used
- → after JDK7
  - ↳ default methods were unlocked
- → after JDK8
  - ↳ static methods were unlocked

```
interface i1 {
        int a = 20;
}
```

by default final, static, public

methods → implicitly abstract and public

→ only method name should be declared

```
interface i2 {
    int b = 20;
    String n = "Sec 1";
    void show();
    default void show1(){...}
    static void show2(){...}
}
```

head

```
void show(){
    System.out.println("...");
}
```

body

interfaces can be implemented by overriding that abstract method.

Static → 'global'

```
class A implements i2 {
    public void show(){
        System.out.println(" Overriden");
    }
}
```

```
class A implements i2 {
    ~~default~~ void show1(){
public  System.out.println("Def");
    }

    public void show2(){
        ~~System.out.println("Static");~~
    }
}
```

Static and default methods are not abstract since they contain a body

static methods cannot be overriden or inherited

One class → any number/multiple interface কে implement করতে পারে

any number of classes → can implement a single interface

interfaces can be inherited by other interfaces.

```
public class Main{
    psvm(String[] args){
        A ob = new A();
        ob.show(); ✓
        ob.show1(); ✓
        ob.show2(); X      will cause error
        i1.show2(); ✓
        i1 r1 = ob;
          r1.show()
          r1.show1()
          r1.show2()
```

object reference demonstrates Runtime Polymorphism and Dynamic Method Dispatch

→ this calls the overriden version

```
        i1. super(). show1();
```

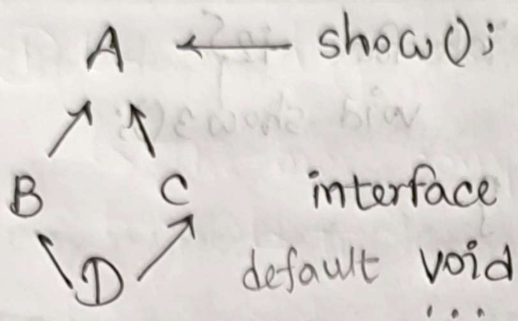
→ this will call the interface version

```
interface i1{                    interface i2{
    void show();                     void show3();
    void show1();                }
}

class A implements i1, i2{
public void   show(){
            System.out.println("——");
    }
public void  show1(){
            ...
    }
public  void  show3(){
            ...
    }
}

public class Main{
  psvm(..){
    A ob =new A();
    i1 r1 = ob;
        r1.show(); ✓
        r1.show1(); ✓
        r1.show3(); ✗ ← doesn't exists in i1

    i2  r2 = ob;
        r2. show3(); ✓ ← exists in i2
}}
```



```
interface A{
default void show(){
    ...}
}
```

```
interface B extends A{
    //empty
}
```

```
interface C extends A{
    //empty
}
```

```
class D implements B,C{
    public void show(){
        System.out.println("...");
    }
}
```

to resolve diamond problem the default/abstract method must be overriden

don't use 'abstract' keyword

```
interface i1 {
        void  show1();
}

interface  i2 {
        void  show2();
}

interface  i3  extends  i1, i2 {
        void  show3();
}

class  A  implements  i3 {
      public  void  show1() {. . . }
      public  void  show2() {. . . }
      public  void  show3 () { . . . }
}
```