SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

23CSB101

OBJECT ORIENTED PROGRAMMING

UNIT II

INHERITANCE, PACKAGES AND INTERFACES

METHOD OVERRIDING

By

M.Kanchana

Assistant Professor/CSE

METHOD OVERRIDING

Method Overriding in Java is when a subclass provides a specific implementation of a method that is already defined in its superclass. The overridden method in the subclass must have the same name, return type, and parameters as the method in the parent class.

Supports Code Reusability

•You **reuse** the superclass method but modify it for specific needs.

METHOD OVERRIDING

```
class Vehicle {
    void speed() {
        System.out.println("Vehicle is moving");
    }
}
class Car extends Vehicle {
    // New method specific to Car
    void speedCar() {
        System.out.println("Car is moving at 60 mph");
    }
}
public class Main {
    public static void main(String[] args) {
        Car myCar = new Car();
        myCar.speed();    // Calls speed() from Vehicle
        myCar.speedCar(); // Calls speedCar() from Car
    }
}
```

METHOD OVERRIDING

- A Car has a more specific behavior compared to a general Vehicle
- Overriding is better if speed() represents the same concept across different vehicle types.
- Using a separate method (speedCar()) is okay if it has a different purpose from speed().

METHOD OVERRIDING

```
class Vehicle {
    void speed() {
        System.out.println("Vehicle is moving");
    }
}
class Car extends Vehicle {
    @Override
    void speed() {
        System.out.println("Car is moving at 60 mph");
    }
}
public class Main {
    public static void main(String[] args) {
        Vehicle myCar = new Car();
            myCar.speed(); // Calls Car's speed() method
    }
}
```

METHOD OVERRIDING

```
class Bank
{
int getRateOfInterest (){
return 0;
}}

class Axis extends Bank {
int getRateOfInterest()
{
return 6;
}}
```

METHOD OVERRIDING

```
class ICICI extends Bank {
int getRateOfInterest

return 15;
}
}
class BankTest
{
public static void main(String[] a)
{
Axis a=new Axis();
 ICICI i=new ICICI();
System.out.println("AXIS: Rate of Interest = "+a.getRateOfInterest());
 System.out.println("ICICI: Rate of Interest = "+i.getRateOfInterest());
}
```

METHOD OVERRIDING

RULES FOR METHOD OVERRIDING:

- The method signature must be same for all overridden methods.
- Instance methods can be overridden only if they are inherited by the subclass.
- A method declared final cannot be overridden.
- A method declared static cannot be overridden but can be re-declared.
- If a method cannot be inherited, then it cannot be overridden.
- Constructors cannot be overridden.

In Java, static methods belong to the class rather than an instance of the class. This means they do not participate in runtime polymorphism **(method overriding**). However, they can be re-declared in a subclass, which is called **method hiding.**

THANK YOU