



SNS COLLEGE OF ENGINEERING

Coimbatore-107
An Autonomous Institution

COURSE NAME : 23CSB201 & Object Oriented Programming

II YEAR/ III SEMESTER

UNIT – II INHERITANCE, PACKAGES, INTERFACE

Topic: Dynamic Method Dispatch

Dr.P.Poonkodi

Assistant Professor(SG)

Department of Computer Science and Technology



Introduction

- Also known as runtime polymorphism
- It is the mechanism by which a call to an overridden method is resolved at runtime rather than at compile time
- This is achieved through method overriding and allows Java to support late binding
- In Java, when a superclass reference variable refers to a subclass object, the method that gets executed is the one in the subclass
- This allows flexibility and extensibility in code



Case Study: Student Performance Evaluation System

- Problem Statement
 - Design a Java program where a superclass **Student** has a method `displayDetails()`. The subclass **GraduateStudent** overrides this method to provide additional details specific to graduate students.



Example

```
class Student
{
// Method to display student details
    public void displayDetails()
    {
        System.out.println("General Student Details");
    }
}
```



Example

```
class GraduateStudent extends Student
{
    // Overriding the displayDetails() method
    @Override
    public void displayDetails()
    {
        System.out.println("Graduate Student Details: Research Area and
                            Thesis Title");
    }
}
```



Example

```
class UndergraduateStudent extends Student
```

```
{
```

```
    // Overriding the displayDetails() method
```

```
    @Override
```

```
    public void displayDetails()
```

```
    {
```

```
        System.out.println("Undergraduate Student Details: Major and  
                             GPA");
```

```
    }
```

```
}
```



public class Main

{

public static void main(String[] args)

{

// Superclass reference to Student object

Student student = new Student();

student.displayDetails();

// Calls Student's method

// Superclass reference to GraduateStudent object

Student gradStudent = new GraduateStudent();

gradStudent.displayDetails();



```
// Calls GraduateStudent's overridden method
// Superclass reference to UndergraduateStudent object
Student undergradStudent = new UndergraduateStudent();
undergradStudent.displayDetails();
// Calls UndergraduateStudent's overridden method
}

}
```



References

- Java : the complete Reference (Eleventh Edition), Herbert Schildt, 2018.

