Inheritance

```java
public class Person {

    private String name;


    // Constructors
    public Person() {

        this.name = "Unknown";

    }


    public Person(String name) {

        this.name = name;

    }


    // Getter and Setter methods
    public void setName(String name) {

        this.name = name;

    }


    public String getName() {

        return name;

    }


    // Method to display information
    public void writeOutput() {

        System.out.println("Name: " + name);

    }
}
```

```java
public class Student extends Person {

    private int studentNumber;


    // Constructors
    public Student() {

        super(); // Calls the no-parameter constructor of Person

        this.studentNumber = 0;

    }


    public Student(String name, int studentNumber) {

        super(name); // Calls the parameterized constructor of Person

        this.studentNumber = studentNumber;

    }


    // Getter and Setter methods
    public void setStudentNumber(int studentNumber) {

        this.studentNumber = studentNumber;

    }


    public int getStudentNumber() {

        return studentNumber;

    }


    // Overridden method
    @Override
    public void writeOutput() {

        super.writeOutput(); // Calls the writeOutput method from Person
```

```java
        System.out.println("Student Number: " + studentNumber);

    }

}


public class Undergraduate extends Student {

    private int level; // 1 for freshman, 2 for sophomore, etc.


    // Constructors
    public Undergraduate() {

        super(); // Calls the no-parameter constructor of Student (and Person)

        this.level = 1; // Default level as freshman

    }


    public Undergraduate(String name, int studentNumber, int level) {

        super(name, studentNumber); // Calls the parameterized constructor of Student (and
Person)

        this.level = level;

    }


    // Getter and Setter methods for level
    public void setLevel(int level) {

        this.level = level;

    }


    public int getLevel() {

        return level;

    }
```

```java
    // Overridden method
    @Override
    public void writeOutput() {
        super.writeOutput(); // Calls the writeOutput method from Student (and Person)
        System.out.println("Level: " + (level == 1 ? "Freshman" : level == 2 ? "Sophomore" : level == 3 ? "Junior" : "Senior"));
    }
}
```

```java
public class InheritanceDemo {
    public static void main(String[] args) {
        // Create and display a Person object
        Person person = new Person("Alice Smith");
        person.writeOutput();
        System.out.println();

        // Create and display a Student object
        Student student = new Student("Bob Johnson", 12345);
        student.writeOutput();
        System.out.println();

        // Create and display an Undergraduate object
        Undergraduate undergrad = new Undergraduate("Charlie Brown", 67890, 3);
        undergrad.writeOutput();
    }
}
```

```
|Name: Alice Smith
|Name: Bob Johnson
|Student Number: 12345
|Name: Charlie Brown
|Student Number: 67890
|Level: Junior
```