

EXP-6.1:

AIM:

Create a base Person class that has properties like name and age, and a method to display basic information. Then, create two subclasses: Student and Teacher, each extending Person. The Student class should include an additional property like grade or course, and the Teacher class should include a property like subject or department. Each subclass should override or extend methods as needed to display complete details. Finally, create instances of both subclasses and demonstrate calling their methods to show how inheritance and method overriding work.

CODE-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Person Class Hierarchy Demo</title>
  <style>
    body {
      font-family: 'Segoe UI', Arial, sans-serif;
      background: linear-gradient(130deg, #e0e7ff 0%, #f2f4ff 35%, #e0f7fa
100%);
      min-height: 100vh;
      margin: 0;
      padding: 0;
      display: flex;
      flex-direction: column;
      align-items: center;
    }
    .container {
      margin-top: 3em;
```

```
padding: 2em 2.5em;
background: #ffffffb2;
border-radius: 18px;
box-shadow: 0 10px 32px rgba(60,60,120,0.12);
width: 100%;
max-width: 420px;
transition: box-shadow .2s;
}
h2 {
text-align: center;
color: #204080;
letter-spacing: 0.5px;
text-shadow: 0 2px 3px #d7d7f7;
margin-top: 0;
}
.btn {
margin: 0.6em 0.7em;
padding: 0.65em 1.4em;
background: linear-gradient(90deg, #6ab7ff 0%, #81c784 100%);
color: #fff;
font-size: 1.07em;
border: none;
border-radius: 8px;
cursor: pointer;
box-shadow: 0 2px 12px rgba(132,165,240,0.13);
transition: background .18s, transform .15s;
outline: none;
}
.btn:hover {
background: linear-gradient(90deg, #42a5f5 0%, #66bb6a 100%);
transform: scale(1.06);
}
.output {
margin-top: 1.5em;
padding: 1.4em;
```

```

border-radius: 12px;
border: 1.5px solid #aeca9c;
background: #f6faff;
color: #204080;
font-size: 1.13em;
min-height: 2.2em;
box-shadow: 0 3px 12px rgba(120,120,160,0.11);
letter-spacing: 0.5px;
display: flex;
align-items: center;
justify-content: center;
transition: box-shadow .2s;
}
.icon {
width: 40px;
height: 40px;
margin-right: 14px;
vertical-align: middle;
}
</style>
</head>
<body>
<div class="container">
<h2>Person Class Hierarchy Demo</h2>
<button class="btn" onclick="showStudent()"><span>👤</span> Show
Student Info</button>
<button class="btn" onclick="showTeacher()"><span>👨🏫</span> Show
Teacher Info</button>
<div class="output" id="output"></div>
</div>

<script>
// Base class: Person
class Person {
constructor(name, age) {

```

```

    this.name = name;
    this.age = age;
  }
  displayInfo() {
    return `Name: ${this.name}, Age: ${this.age}`;
  }
}

// Subclass: Student
class Student extends Person {
  constructor(name, age, course) {
    super(name, age);
    this.course = course;
  }
  displayInfo() {
    return `👤 Name: ${this.name}, Age: ${this.age}, Course:
${this.course}`;
  }
}

// Subclass: Teacher
class Teacher extends Person {
  constructor(name, age, subject) {
    super(name, age);
    this.subject = subject;
  }
  displayInfo() {
    return `👤 Name: ${this.name}, Age: ${this.age}, Subject:
${this.subject}`;
  }
}

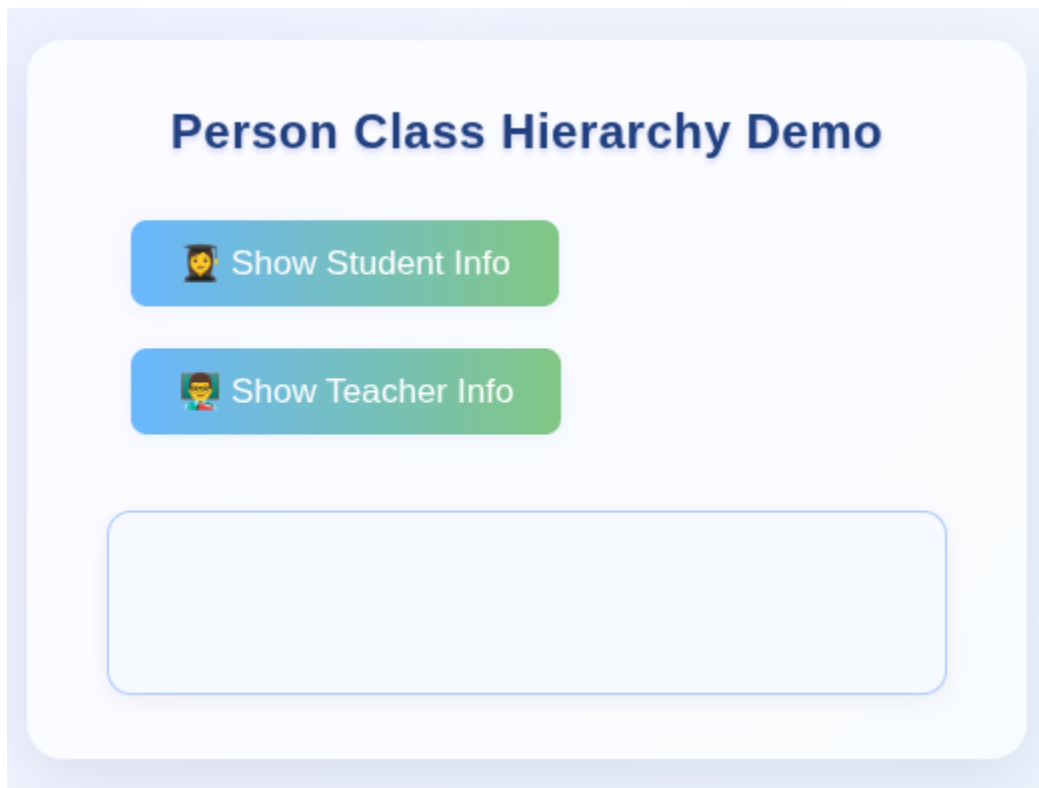
// Sample instances
const student1 = new Student("Alice", 20, "Mathematics");
const teacher1 = new Teacher("Mr. Smith", 45, "Physics");

// UI functions
function showStudent() {

```

```
        document.getElementById("output").textContent =
student1.displayInfo();
    }
    function showTeacher() {
        document.getElementById("output").textContent =
teacher1.displayInfo();
    }
</script>
</body>
</html>
```

OUTPUT:



Person Class Hierarchy Demo



Show Student Info



Show Teacher Info



Name: Alice, Age: 20, Course:
Mathematics

Person Class Hierarchy Demo



Show Student Info



Show Teacher Info



Name: Mr. Smith, Age: 45, Subject:
Physics