**Practical 4: Student data**  
  
// SPDX-License-Identifier: Bhide License

pragma solidity ^0.8.0;

contract StudentRegistry {

    // Define a structure to hold student details

    struct Student {

        string name;  // Name of the student

        uint256 age;  // Age of the student

    }

    // Array to store student records

    Student[] private students;

    // Event to log when a student is added

    event StudentAdded(string name, uint256 age);

    // Event to log the received Ether value

    event ReceivedEther(address indexed sender, uint256 value);

    // Function to receive Ether directly

    receive() external payable {

        // Log the received Ether value

        emit ReceivedEther(msg.sender, msg.value);

    }

    // Fallback function to receive Ether and log it

    fallback() external payable {

        emit ReceivedEther(msg.sender, msg.value);

    }

    // Function to add a new student to the registry

    function addStudent(string memory name, uint256 age) public {

        students.push(Student(name, age));

        emit StudentAdded(name, age);  // Emit event for added student

    }

    // Function to retrieve a student's details by index

    function getStudent(uint256 index) public view returns (string memory, uint256) {

        require(index < students.length, "Student not found");

        return (students[index].name, students[index].age);

    }

    // Function to get the total count of students in the registry

    function getStudentCount() public view returns (uint256) {

        return students.length;

    }

}