Student Solidity

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract StudentRegistry {
    // Structure to hold student data
    struct Student {
        uint id;
        string name;
        uint age;
        string course;
    }

    // Array to hold the list of students
    Student[] public students;
    // Mapping to track the number of students
    uint public studentCount;

    // Fallback function to handle calls with data or unrecognized functions
    fallback() external payable {
        // This function can be used to handle unexpected calls
    }

    // Receive function to handle plain Ether transfers
    receive() external payable {
        // This function can be used to accept donations or track incoming Ether
    }

    // Function to add a new student
    function addStudent(string memory _name, uint _age, string memory _course) public {
        studentCount++;
        students.push(Student(studentCount, _name, _age, _course));
    }

    // Function to retrieve a student by ID
    function getStudent(uint _id) public view returns (uint, string memory, uint, string memory) {
        require(_id > 0 && _id <= studentCount, "Student not found.");
        Student memory student = students[_id - 1]; // Arrays are 0-indexed
        return (student.id, student.name, student.age, student.course);
    }

    // Function to get the total number of students
    function getTotalStudents() public view returns (uint) {
        return studentCount;
    }
}
```