BT bank

```solidity
pragma solidity 0.9.0;

contract Bank
{
    int bal;

    constructor() public {
        bal = 1;
    }

    function getBalance() view public returns(int){
        return bal;
    }

    function deposit(int amt) public  {
        bal=bal+amt;
    }

    function withdraw(int amt) public {
        bal=bal-amt;
    }

}
```

BT student

```solidity
pragma solidity ^0.8.0;

contract StudentData {

    struct Student {
        uint id;
        string name;
        uint age;
    }

    Student[] public students;

    event NewStudent(uint id, string name, uint age);

    function createStudent(uint id, string memory name, uint age) public {
        students.push(Student(id, name, age));
        emit NewStudent(id, name, age);
    }

    function getStudent(uint index) public view returns (uint, string memory, uint) {
        require(index < students.length, "Invalid index");
        Student memory s = students[index];
        return (s.id, s.name, s.age);
    }

    fallback() external {
        revert("Fallback function called. Please use proper functions.");
    }
}
```