

```

pragma solidity ^0.4.18;
// SPDX-License-Identifier: MIT
contract BCDV1011_Lab_1 {

    // Setting the variable 'owner' to the address of the contract deployer.

    address owner;
    constructor() public {
        owner = msg.sender;
    }

    // Create a modifier to ensure the current address interacting with a
    // particular function, is the same as the contract owner(deployer).
    modifier onlyOwner{
        require(msg.sender == owner);
        _;
    }

    // Create another modifier to ensure a student mark is between 0 and 100.
    modifier markInRange(uint mark){
        require(mark >= 0 && mark <= 100);
        _;
    }

    // This struct will hold student data.
    // Structs allow the grouping of many variables of multiple types
    // into one user defined type.

    struct Student {
        string name;
        string subject;
        uint8 mark;
    }

    event StudentAdded(
        string content
    );

    // Here we create a mapping of key/value pairs.
    // An address is mapped to a Student struct.
    // This mapping is called 'students'.

    mapping (address => Student) students;

```

```

    // Mappings in Solidity are not iterable, and don't have length or count
properties.
    // Solution: Keep track of the size manually by declaring a state variable
    // and increase/decrease this variable whenever you have a push/delete
    // operation on the mapping.

    uint mapSize;

    //
    //Function to add values to the mapping.
    //
    // Insert a modifier to ensure student mark is between 0 and 100.
    function adding_values(address _address, string memory _name, string memory
_subject, uint8 _mark) public markInRange(_mark) {

        Student storage student = students[_address];
        student.name = _name;
        student.subject = _subject;
        student.mark = _mark;
        // increase the map size by +1 for the new student.
        mapSize++;
        //
        emit StudentAdded("Student added successfully.");

    }

    //
    // Function to retrieve student info from the mapping.
    //
    // Insert a modifier so that only the contract owner can call this function.
    function get_student_info(address _address) view public onlyOwner returns
(string memory, string memory, uint8) {
        string memory _name = students[_address].name;
        string memory _subject = students[_address].subject;
        uint8 _mark = students[_address].mark;
        return (_name, _subject, _mark);
    }

    //
    // Function to count number of students.
    //
    function count_students() view public returns (uint) {
        return mapSize;
    }

```

}