#include <iostream>

using namespace std;

class Course

{

public:

// Constructor

Course(const string& courseName, int capacity);

// Deep copy constructor

Course(const Course& other);

// Destructor

~Course();

// Getter for course name

string getCourseName() const;

// Add student to the course

void addStudent(const string& name);

// Drop student from the course

void dropStudent(const string& name);

// Getter for array of students

string\* getStudents() const;

// Getter for number of students

int getNumberOfStudents() const;

private:

string courseName;

string\* students;

int numberOfStudents;

int capacity;

};

// Constructor

Course::Course(const string& courseName, int capacity)

{

numberOfStudents = 0;

this->courseName = courseName;

this->capacity = capacity;

students = new string[capacity];

}

// Deep copy constructor

Course::Course(const Course& other)

{

numberOfStudents = other.numberOfStudents;

courseName = other.courseName;

capacity = other.capacity;

students = new string[capacity];

for (int i = 0; i < numberOfStudents; ++i)

{

students[i] = other.students[i];

}

}

// Destructor

Course::~Course()

{

delete[] students;

}

// Getter for course name

string Course::getCourseName() const

{

return courseName;

}

// Add student to the course

void Course::addStudent(const string& name)

{

students[numberOfStudents] = name;

numberOfStudents++;

}

// Drop a student from the course

void Course::dropStudent(const string& name)

{

// Find the student

int position = -1;

for (int i = 0; i < numberOfStudents; ++i)

{

if (students[i] == name)

{

position = i;

break;

}

}

// If found, remove the student

if (position != -1)

{

for (int i = position; i < numberOfStudents - 1; ++i)

{

students[i] = students[i + 1];

}

numberOfStudents--;

}

}

string\* Course::getStudents() const

{

return students;

}

int Course::getNumberOfStudents() const

{

return numberOfStudents;

}

int main() {

// Create courses

Course course1("oops", 20);

Course course2("Discrete Structure", 10);

// Adding students to course1

course1.addStudent("bilal");

course1.addStudent("shahzad");

course1.addStudent("aziz");

// Adding students to course2

course2.addStudent("saad");

course2.addStudent("anill");

// Display the number of students in course1

cout << "Number of students in course1: " << course1.getNumberOfStudents() << "\n";

// Display the names of students in course1

string\* students = course1.getStudents();

for (int i = 0; i < course1.getNumberOfStudents(); i++)

cout << students[i] << ", ";

cout << endl;

// deep copy of course1

Course course3(course1);

// Drop student from course3

course3.dropStudent("bilal");

cout << "Number of students in course 3 after dropping bilal: " << course3.getNumberOfStudents() << "\n";

students = course3.getStudents();

for (int i = 0; i < course3.getNumberOfStudents(); i++)

cout << students[i] << ", ";

cout << "\n";

return 0;

}