Question:1

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;
class Student
{
private:
    string name;
public:
    Student(const string& name) : name(name) {}
    string getName() const
    {
        return name;
    }
};
class Course
{
public:
    Course(const string& name) : name(name) {}
    // Deep copy constructor
    Course(const Course& other) : name(other.name)
    {
        students.reserve(other.students.size());
        for (const Student& student : other.students)
        {
            students.push_back(student);
        }
    }
    void addStudent(const string& studentName)
    {
        Student student(studentName);
        students.push_back(student);
    }
    void dropStudent(const string& studentName)
    {
        for (auto it = students.begin(); it != students.end(); ++it)
        {
            if (it->getName() == studentName)
            {
                students.erase(it);
                return;
            }
        }
    }
    void printStudents()
    {
        cout << "Students in " << name << ":\n";
        for (const Student& student : students)
        {
            cout << student.getName() << "\n";
        }
    }
private:
    string name;
    vector<Student> students;
};
```

```cpp
int main()
{
    Course math("Math");
    math.addStudent("Sara");
    math.addStudent("Bisma");
    math.addStudent("Charlie");
    Course copyMath = math;
    math.dropStudent("Bisma");
    cout << "Original Math Course:\n";
    math.printStudents();
    cout << "\nCopied Math Course:\n";
    copyMath.printStudents();
    return 0;
}
```