```cpp
#include <iostream>
using namespace std;

/*class Box {
private:
    int x;
    int y;
public:
    Box()
    {
        x = 0;
        y = 0;
    }
    void setX(int a)
    {
        x = a;
    }
    void setY(int a)
    {
        y = a;
    }
    int getX()
    {
        return x;
    }
    int getY()
    {
        return y;
    }
};
int main()
{
    Box a, b;
    a.setX(10);
    a.setY(20);
    b = a;
    cout << "After assignment statement" << endl;
    cout << a.getX() << " " << a.getY() << endl;
    cout << b.getX() << " " << b.getY() << endl;
    a.setX(30);
    a.setY(40);
    b.setX(50);
    b.setY(60);
    cout << "After updating a and b" << endl;
    cout << a.getX() << " " << a.getY() << endl;
    cout << b.getX() << " " << b.getY() << endl;
    return 0;
}*/

```

```
class StudentTestScores
{
private:
    string studentName;  // The student's name
    double* testScores;  // Points to array of test scores
    int numTestScores;   // Number of test scores
public:
    StudentTestScores(int size)
    {
        studentName = " ";
        numTestScores = size;
        testScores = new double[size];
        for (int i = 0; i < size; i++)
            testScores[i] = 0;
    }

    /*~StudentTestScores()
    {
        delete[] testScores;
    }*/

    void setTestScore(double score, int index)
    {
        testScores[index] = score;
    }

    void setStudentName(string name)
    {
        studentName = name;
    }

    string getStudentName() const
    {
        return studentName;
    }

    int getNumTestScores()
    {
        return numTestScores;
    }

    double getTestScore(int index) const
    {
        return testScores[index];
    }
    void displayStudent()
    {
        cout << "Name: " << studentName << endl;
        cout << "Test Scores: ";
        for (int i = 0; i < numTestScores; i++)
             cout << testScores[i] << " ";
        cout << endl;
    }

    // Overloaded = operator
    const StudentTestScores operator=(const StudentTestScores& right)
    {
        delete[] testScores;
        studentName = right.studentName;
        numTestScores = right.numTestScores;
        testScores = new double[numTestScores];
        for (int i = 0; i < numTestScores; i++)
            testScores[i] = right.testScores[i];
        return *this;
    }

    // Copy constructor
    StudentTestScores(const StudentTestScores& obj)
    {
        studentName = obj.studentName;
        numTestScores = obj.numTestScores;
        testScores = new double[numTestScores];
        for (int i = 0; i < numTestScores; i++)
            testScores[i] = obj.testScores[i];
    }
};

int main()
{
    StudentTestScores student1(3);
    student1.setStudentName("Jack");
    student1.setTestScore(100.0, 0);
    student1.setTestScore(95.0, 1);
    student1.setTestScore(80, 2);
    student1.displayStudent();

    StudentTestScores student2(3);
    student2 = student1;
    student2.setStudentName("David");
    student2.displayStudent();

    StudentTestScores student3 = student1;
    student3.setStudentName("Adam");
    student3.displayStudent();

    student1.setTestScore(88, 0);
    student2.setTestScore(77, 1);
    student3.setTestScore(66, 2);
    cout << "After updating students 1,2, and 3" << endl;
    student1.displayStudent();
    student2.displayStudent();
    student3.displayStudent();

    return 0;
}
```