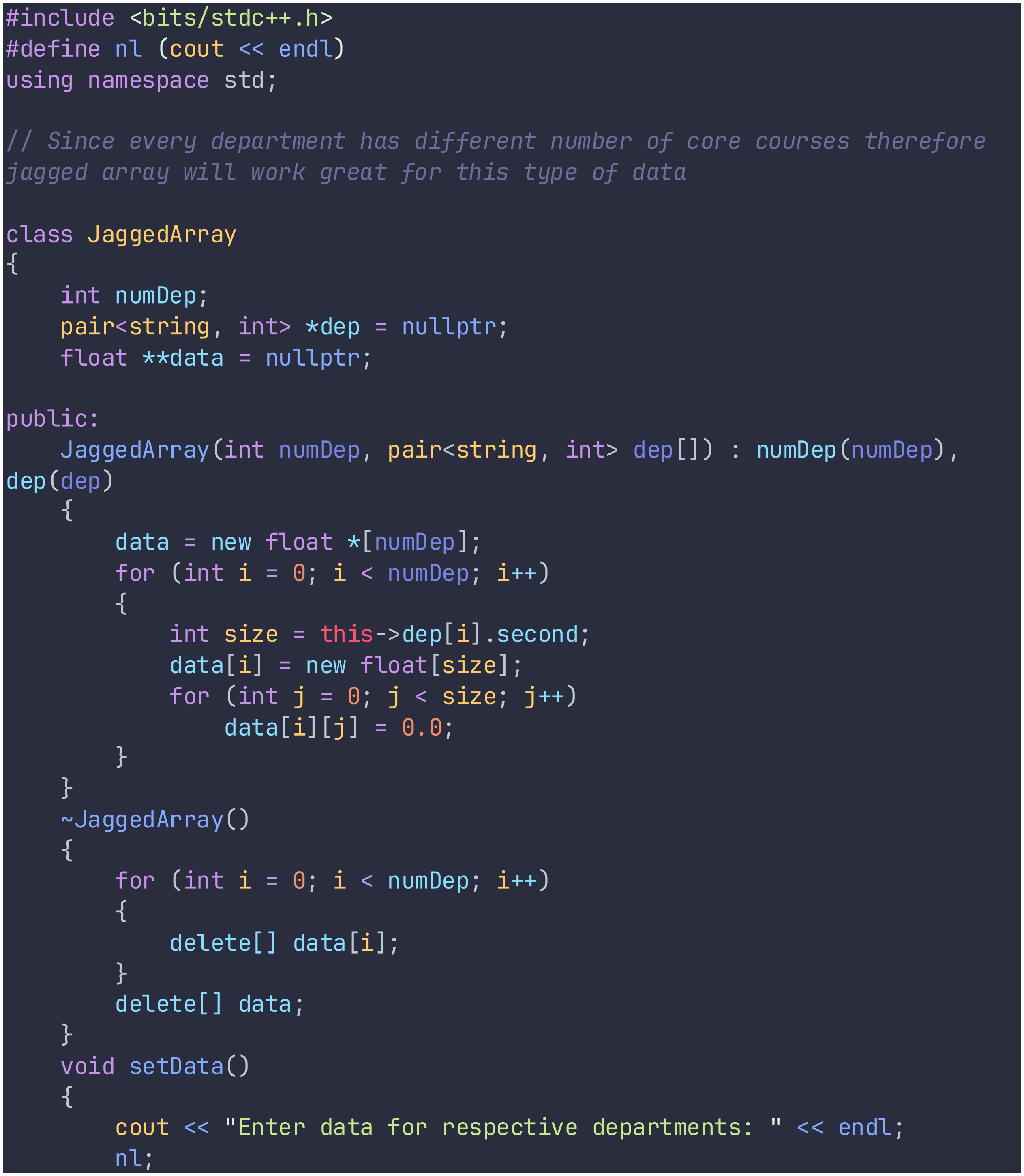
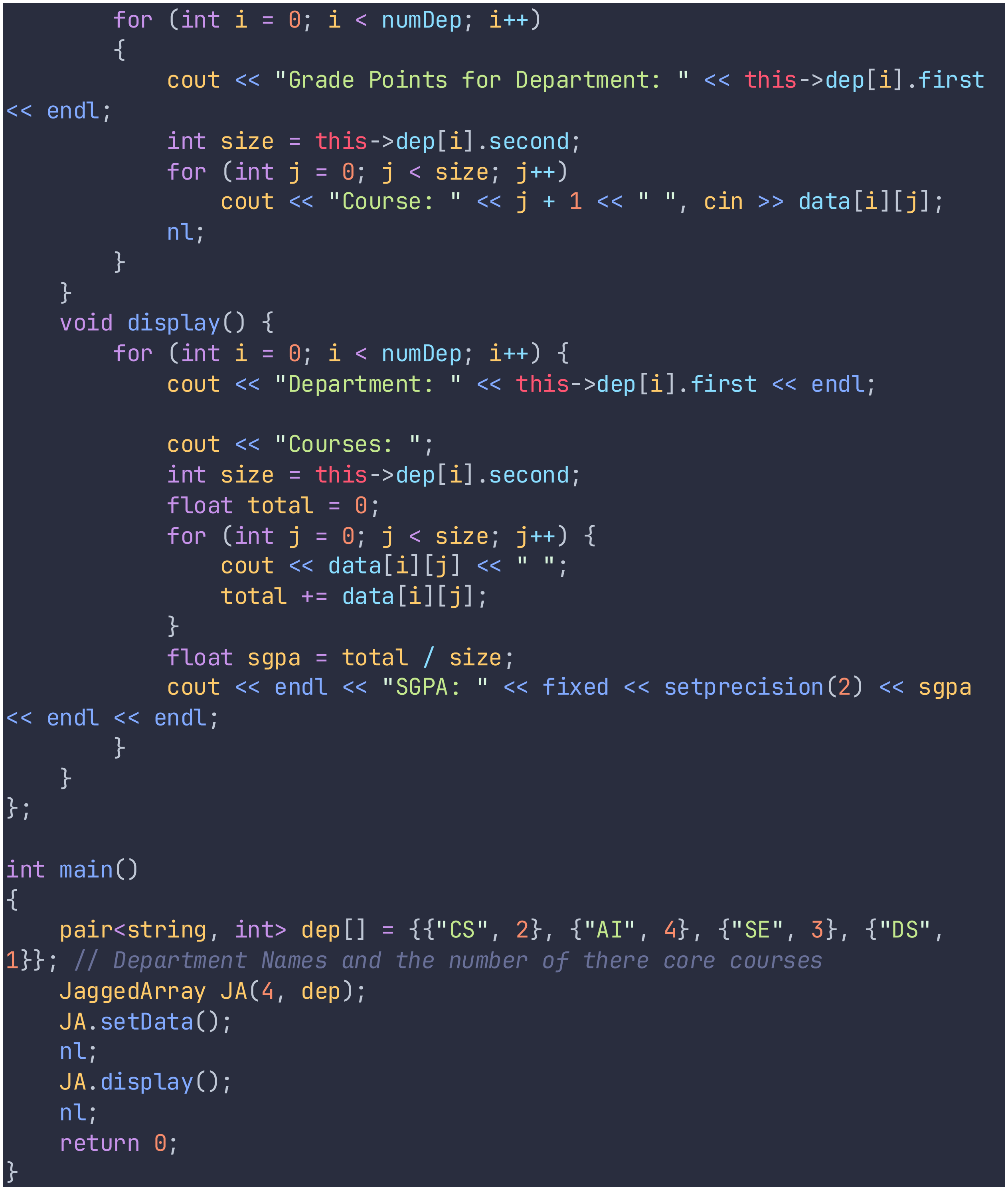
DS LAB 02

Question 5

Code:

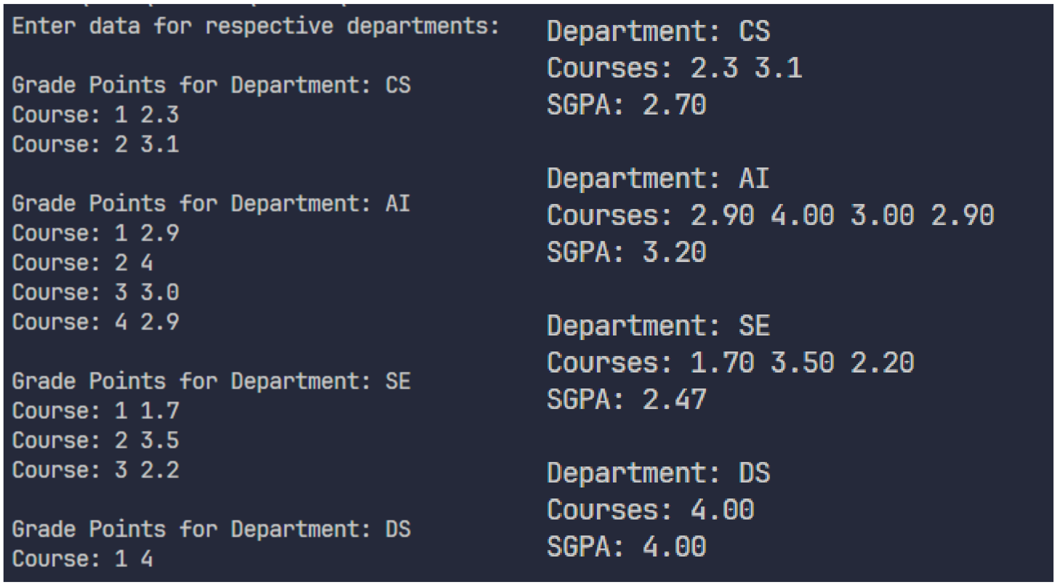


```
#include <bits/stdc++.h>
#define nl (cout << endl)
using namespace std;

// Since every department has different number of core courses therefore
jagged array will work great for this type of data

class JaggedArray
{
    int numDep;
    pair<string, int> *dep = nullptr;
    float **data = nullptr;

public:
    JaggedArray(int numDep, pair<string, int> dep[]) : numDep(numDep),
dep(dep)
    {
        data = new float *[numDep];
        for (int i = 0; i < numDep; i++)
        {
            int size = this->dep[i].second;
            data[i] = new float[size];
            for (int j = 0; j < size; j++)
                data[i][j] = 0.0;
        }
    }
    ~JaggedArray()
    {
        for (int i = 0; i < numDep; i++)
        {
            delete[] data[i];
        }
        delete[] data;
    }
    void setData()
    {
        cout << "Enter data for respective departments: " << endl;
        nl;
```



```
        for (int i = 0; i < numDep; i++)
        {
            cout << "Grade Points for Department: " << this->dep[i].first
<< endl;
            int size = this->dep[i].second;
            for (int j = 0; j < size; j++)
                cout << "Course: " << j + 1 << " ", cin >> data[i][j];
            nl;
        }
    }
    void display() {
        for (int i = 0; i < numDep; i++) {
            cout << "Department: " << this->dep[i].first << endl;

            cout << "Courses: ";
            int size = this->dep[i].second;
            float total = 0;
            for (int j = 0; j < size; j++) {
                cout << data[i][j] << " ";
                total += data[i][j];
            }
            float sgpa = total / size;
            cout << endl << "SGPA: " << fixed << setprecision(2) << sgpa
<< endl << endl;
        }
    }
};

int main()
{
    pair<string, int> dep[] = {{"CS", 2}, {"AI", 4}, {"SE", 3}, {"DS",
1}}; // Department Names and the number of there core courses
    JaggedArray JA(4, dep);
    JA.setData();
    nl;
    JA.display();
    nl;
    return 0;
}
```

Output:

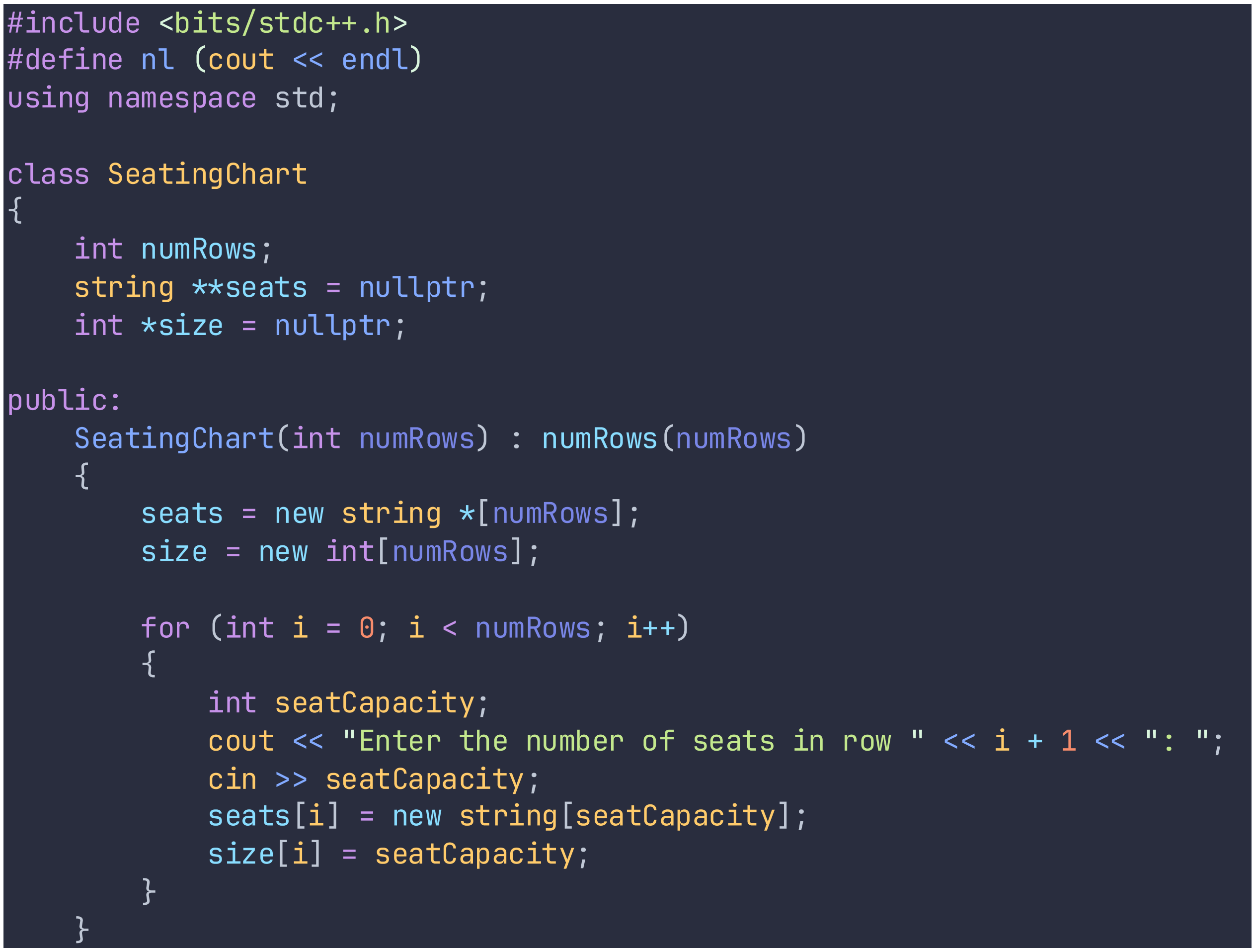


```
Enter data for respective departments:

Grade Points for Department: CS
Course: 1 2.3
Course: 2 3.1

Grade Points for Department: AI
Course: 1 2.9
Course: 2 4
Course: 3 3.0
Course: 4 2.9

Grade Points for Department: SE
Course: 1 1.7
Course: 2 3.5
Course: 3 2.2

Grade Points for Department: DS
Course: 1 4
```

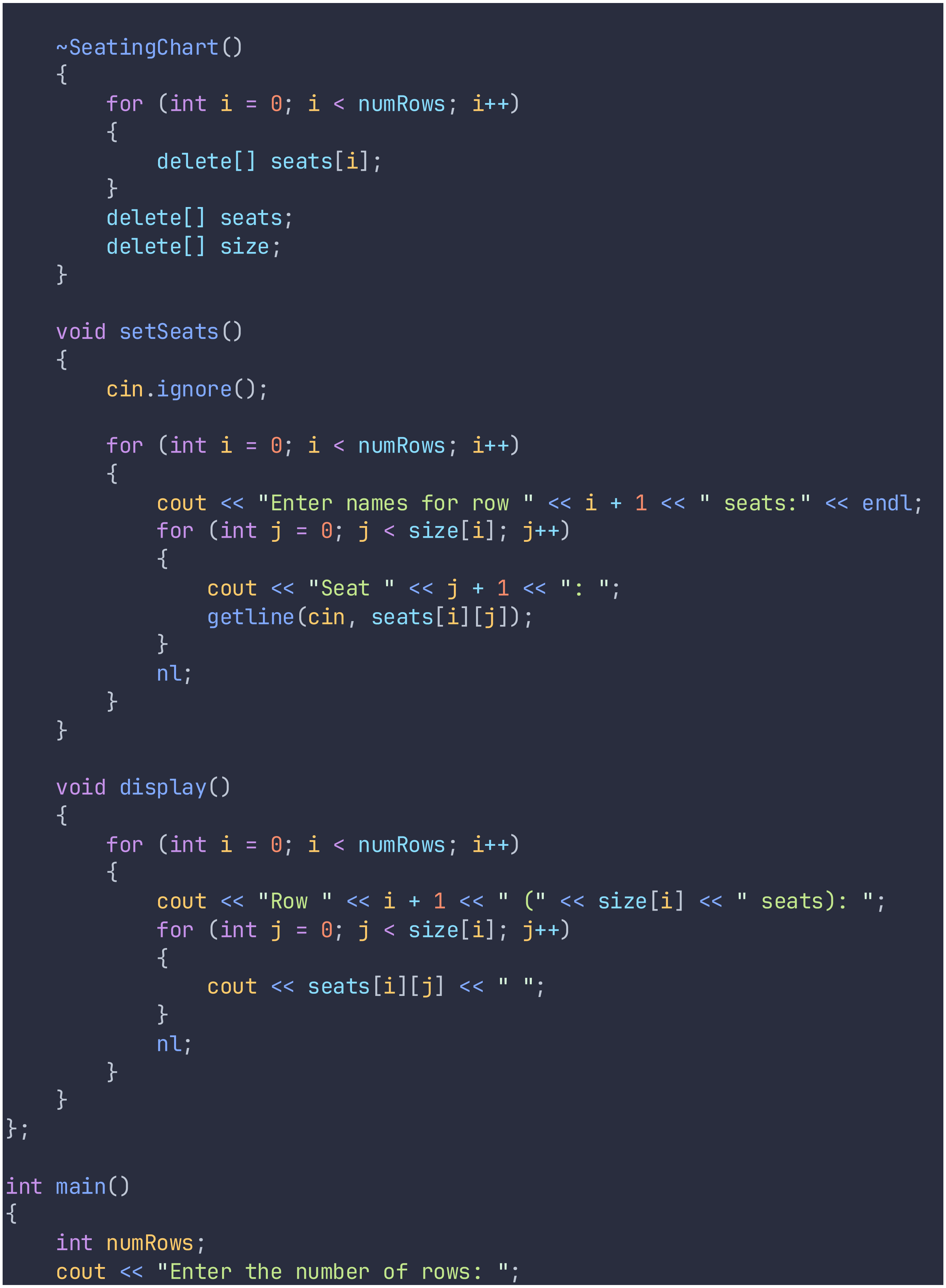
```
Department: CS
Courses: 2.3 3.1
SGPA: 2.70

Department: AI
Courses: 2.90 4.00 3.00 2.90
SGPA: 3.20

Department: SE
Courses: 1.70 3.50 2.20
SGPA: 2.47

Department: DS
Courses: 4.00
SGPA: 4.00
```

Question 6

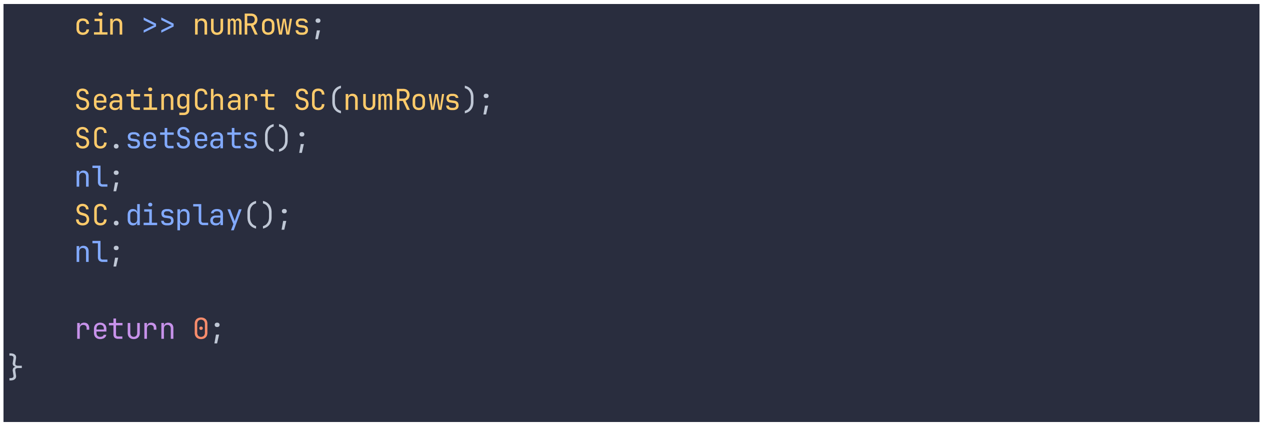
Code:



```cpp
#include <bits/stdc++.h>
#define nl (cout << endl)
using namespace std;

class SeatingChart
{
    int numRows;
    string **seats = nullptr;
    int *size = nullptr;

public:
    SeatingChart(int numRows) : numRows(numRows)
    {
        seats = new string *[numRows];
        size = new int[numRows];

        for (int i = 0; i < numRows; i++)
        {
            int seatCapacity;
            cout << "Enter the number of seats in row " << i + 1 << ": ";
            cin >> seatCapacity;
            seats[i] = new string[seatCapacity];
            size[i] = seatCapacity;
        }
    }
```

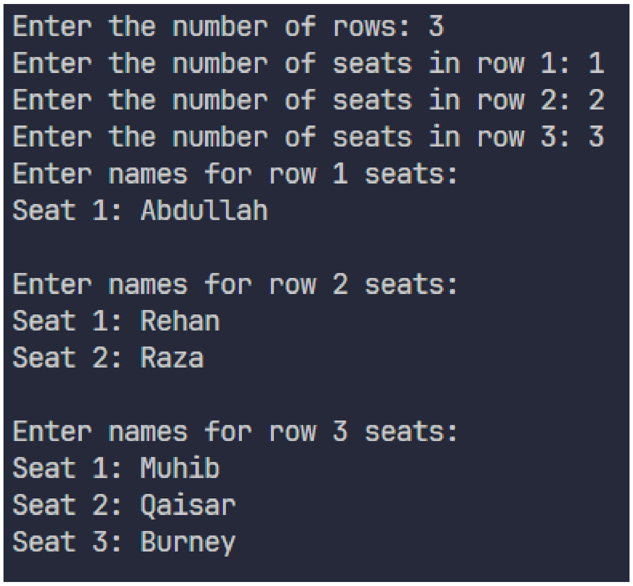


```cpp
    ~SeatingChart()
    {
        for (int i = 0; i < numRows; i++)
        {
            delete[] seats[i];
        }
        delete[] seats;
        delete[] size;
    }

    void setSeats()
    {
        cin.ignore();

        for (int i = 0; i < numRows; i++)
        {
            cout << "Enter names for row " << i + 1 << " seats:" << endl;
            for (int j = 0; j < size[i]; j++)
            {
                cout << "Seat " << j + 1 << ": ";
                getline(cin, seats[i][j]);
            }
            nl;
        }
    }

    void display()
    {
        for (int i = 0; i < numRows; i++)
        {
            cout << "Row " << i + 1 << " (" << size[i] << " seats): ";
            for (int j = 0; j < size[i]; j++)
            {
                cout << seats[i][j] << " ";
            }
            nl;
        }
    }
};

int main()
{
    int numRows;
    cout << "Enter the number of rows: ";
```

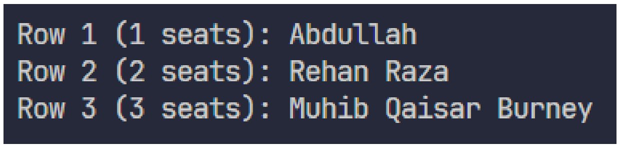


```
    cin >> numRows;

    SeatingChart SC(numRows);
    SC.setSeats();
    nl;
    SC.display();
    nl;

    return 0;
}
```

Output:



```
Enter the number of rows: 3
Enter the number of seats in row 1: 1
Enter the number of seats in row 2: 2
Enter the number of seats in row 3: 3
Enter names for row 1 seats:
Seat 1: Abdullah

Enter names for row 2 seats:
Seat 1: Rehan
Seat 2: Raza

Enter names for row 3 seats:
Seat 1: Muhib
Seat 2: Qaisar
Seat 3: Burney
```



```
Row 1 (1 seats): Abdullah
Row 2 (2 seats): Rehan Raza
Row 3 (3 seats): Muhib Qaisar Burney
```