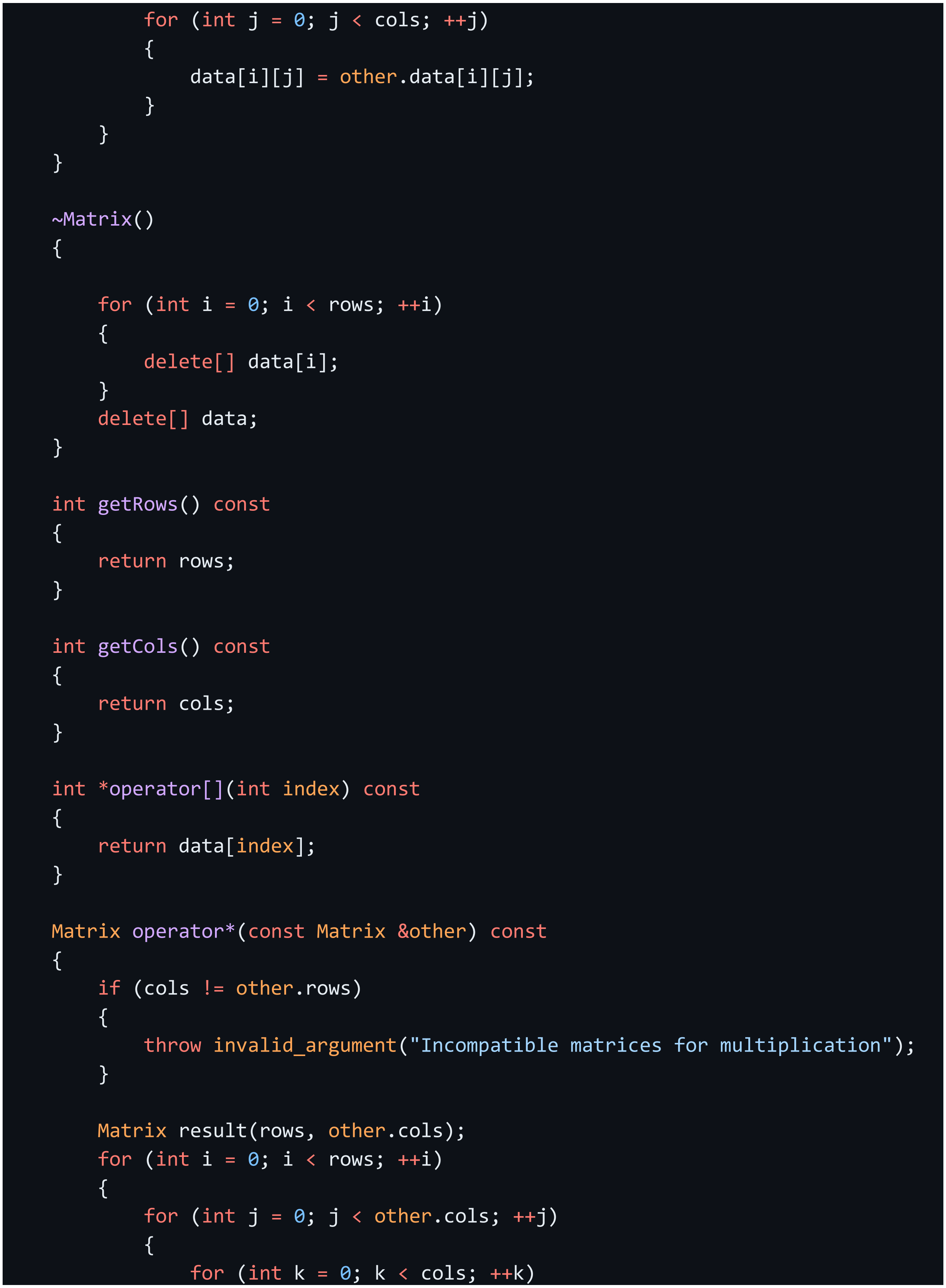
LAB SESSION 13

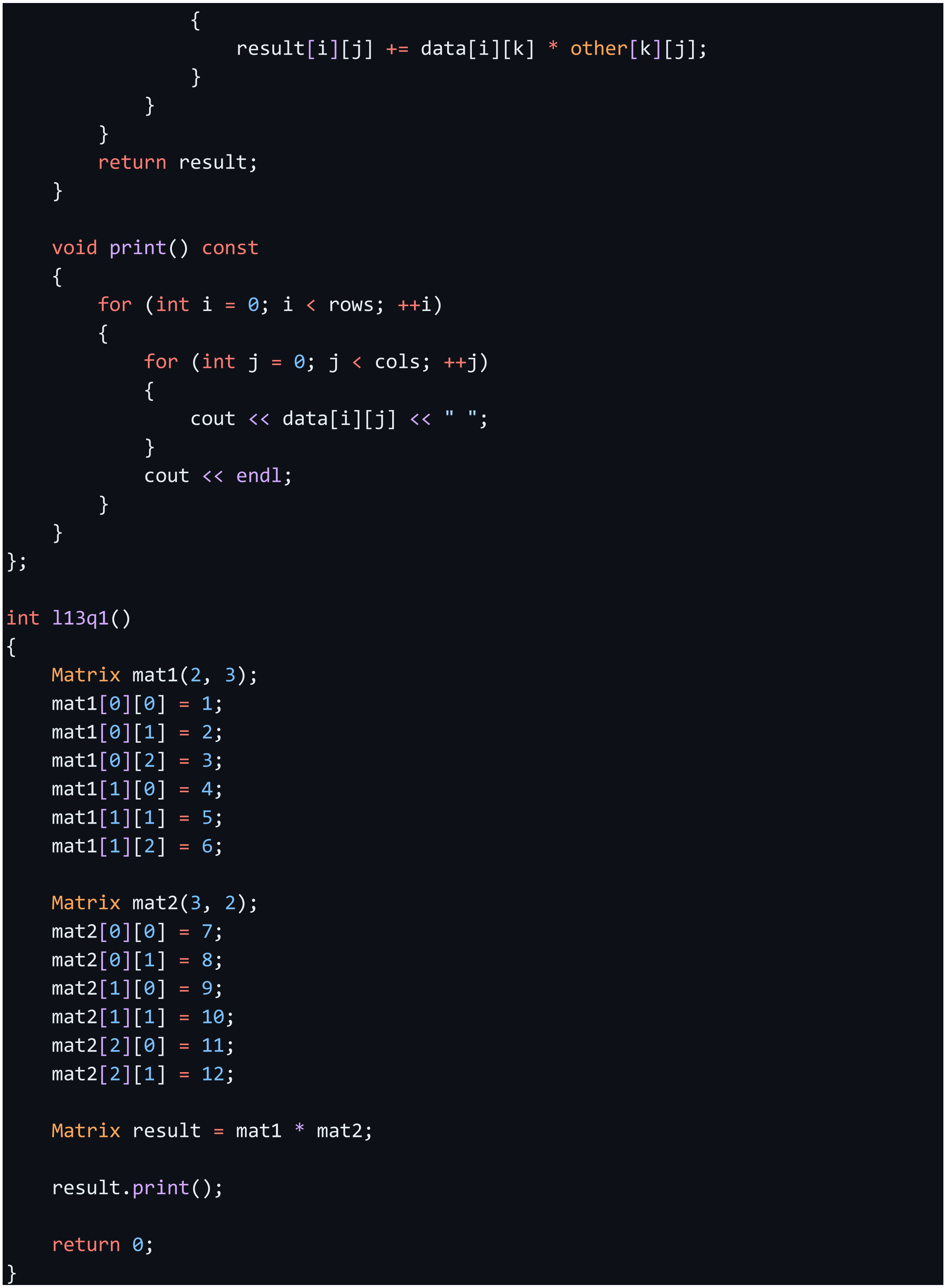
1. Overload the * operator to multiply two Matrix objects (define a simple Matrix class with a 2D array).

Code:

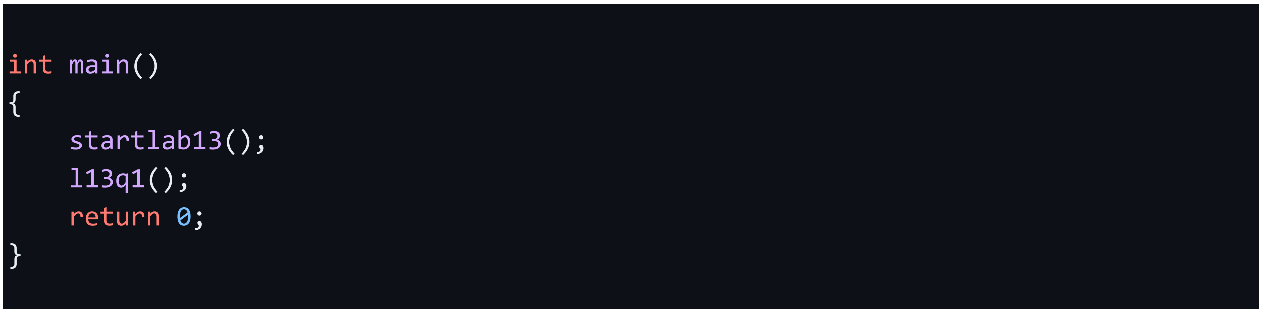
```
#include <iostream>
using namespace std;

int startlab13()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Start of Lab 13" << endl;
    return 0;
}
class Matrix
{
private:
    int rows, cols;
    int **data;

public:
    Matrix(int rows, int cols) : rows(rows), cols(cols)
    {

        data = new int *[rows];
        for (int i = 0; i < rows; ++i)
        {
            data[i] = new int[cols];

            for (int j = 0; j < cols; ++j)
            {
                data[i][j] = 0;
            }
        }
    }

    Matrix(const Matrix &other) : rows(other.rows), cols(other.cols)
    {

        data = new int *[rows];
        for (int i = 0; i < rows; ++i)
        {
            data[i] = new int[cols];
```



```
            for (int j = 0; j < cols; ++j)
            {
                data[i][j] = other.data[i][j];
            }
        }
    }

    ~Matrix()
    {

        for (int i = 0; i < rows; ++i)
        {
            delete[] data[i];
        }
        delete[] data;
    }

    int getRows() const
    {
        return rows;
    }

    int getCols() const
    {
        return cols;
    }

    int *operator[](int index) const
    {
        return data[index];
    }

    Matrix operator*(const Matrix &other) const
    {
        if (cols != other.rows)
        {
            throw invalid_argument("Incompatible matrices for multiplication");
        }

        Matrix result(rows, other.cols);
        for (int i = 0; i < rows; ++i)
        {
            for (int j = 0; j < other.cols; ++j)
            {
                for (int k = 0; k < cols; ++k)
```

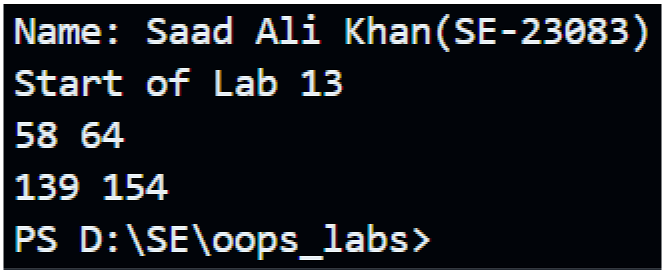


```
                {
                    result[i][j] += data[i][k] * other[k][j];
                }
            }
        }
        return result;
    }

    void print() const
    {
        for (int i = 0; i < rows; ++i)
        {
            for (int j = 0; j < cols; ++j)
            {
                cout << data[i][j] << " ";
            }
            cout << endl;
        }
    }
};

int l13q1()
{
    Matrix mat1(2, 3);
    mat1[0][0] = 1;
    mat1[0][1] = 2;
    mat1[0][2] = 3;
    mat1[1][0] = 4;
    mat1[1][1] = 5;
    mat1[1][2] = 6;

    Matrix mat2(3, 2);
    mat2[0][0] = 7;
    mat2[0][1] = 8;
    mat2[1][0] = 9;
    mat2[1][1] = 10;
    mat2[2][0] = 11;
    mat2[2][1] = 12;

    Matrix result = mat1 * mat2;

    result.print();

    return 0;
}
```



```
int main()
{
    startlab13();
    l13q1();
    return 0;
}
```

Output:



```
Name: Saad Ali Khan(SE-23083)
Start of Lab 13
58 64
139 154
PS D:\SE\oops_labs>
```