Problem 2: Matrix Template Class

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

template <typename T, int Rows, int Cols>
class Matrix {
private:
    T data[Rows][Cols];
public:
    Matrix() {
        for (int i = 0; i < Rows; ++i)
            for (int j = 0; j < Cols; ++j)
                data[i][j] = T();
    }

    T& at(int row, int col) {
        if (row < 0 || row >= Rows || col < 0 || col >= Cols)
            throw out_of_range("Invalid index");
        return data[row][col];
    }

    Matrix operator+(const Matrix& other) const {
        Matrix result;
        for (int i = 0; i < Rows; ++i)
```

```
            for (int j = 0; j < Cols; ++j)
                result.data[i][j] = this->data[i][j] + other.data[i][j];
        return result;
    }

    void out() const {
        for (int i = 0; i < Rows; ++i) {
            for (int j = 0; j < Cols; ++j)
                cout << data[i][j] << " ";
            cout << endl;
        }
    }
};

int main() {
    Matrix<int, 2, 2> mat1;
    Matrix<int, 2, 2> mat2;

    try {
        mat1.at(0, 0) = 1; mat1.at(0, 1) = 2;
        mat2.at(1, 0) = 3; mat2.at(1, 1) = 4;

        Matrix<int, 2, 2> sum = mat1 + mat2;

        cout << "Matrix Addition Result:" << endl;
        sum.out();
```

```
        mat1.at(1, 2) = 3;
    } catch (exception &e) {
        cout << e.what();
    }

    return 0;
}
```