

VIT - Vellore

Name: DIVYANSHU SINGH .

Email: divyanshu.singh2024a@vitstudent.ac.in

Roll no: 24BCT0101

Phone: 9999999999

Branch: PRIYADHARSINI M_OOPS

Department: admin

Batch: VL2024250502354

Degree: admin

Scan to verify results



BCSE102P_Structured and Object Oriented Programming Lab_VL2024250502354

VIT V_Structured and OOP_Lab 5_COD_Medium_Friend Functions and Friend Classes

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Implement a Matrix class that represents a two-dimensional matrix of integers. The class should have methods to create a matrix, set its elements, and print the matrix. Additionally, implement a function to calculate the transpose of a matrix. The program should prompt the user to enter the dimensions of the matrix and its elements, display the original matrix, and then display the transposed matrix.

Function Signature: Matrix Transpose(const Matrix& matrix);

Answer

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

class Matrix {
public:
    Matrix(int rows, int cols) : numRows(rows), numCols(cols), data(rows,
vector<int>(cols)) {}
```

```
    int numRows;
    int numCols;
    vector<vector<int>> data;
```

```
    void setElement(int row, int col, int value) {
        data[row][col] = value;
    }
```

```
    int getElement(int row, int col) const {
        return data[row][col];
    }
```

```
    void print() const {
        for (int i = 0; i < numRows; ++i) {
            for (int j = 0; j < numCols; ++j) {
                cout << data[i][j] << (j == numCols - 1 ? "" : " ");
            }
            cout << endl;
        }
    };
```

```
    Matrix Transpose(const Matrix& matrix) {
        Matrix transposed(matrix.numCols, matrix.numRows);
        for (int i = 0; i < matrix.numRows; ++i) {
            for (int j = 0; j < matrix.numCols; ++j) {
                transposed.setElement(j, i, matrix.getElement(i, j));
            }
        }
        return transposed;
    }
```

```
int main() {
```

```
int numRows, numCols;
cin >> numRows >> numCols;

Matrix matrix(numRows, numCols);
for (int i = 0; i < numRows; ++i) {
    for (int j = 0; j < numCols; ++j) {
        int value;
        cin >> value;
        matrix.setElement(i, j, value);
    }
}

cout << "Original Matrix:" << endl;
matrix.print();

Matrix transposedMatrix = Transpose(matrix);
cout << "Transposed Matrix:" << endl;
transposedMatrix.print();

return 0;
}
```

Status : Correct

Marks : 10/10