

## VIT - Vellore

Name: RONIT MEXSON .

Email: ronit.mexson2024@vitstudent.ac.in

Roll no: 24BAI0036

Phone: 9999999999

Branch: ARUMUGA ARUN R\_OOPS

Department: admin

Batch: VL2024250502365

Degree: admin

Scan to verify results



### BCSE102P\_Structured and Object Oriented Programming Lab\_VL2024250502365

#### 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**

// You are using GCC

```

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

class Matrix{
    int row, col;
    vector<vector<int>> data;
public:
    Matrix(int r, int c) : row(r), col(c), data(r, vector<int>(c)) {}

    void setElements(){
        for (int i = 0; i < row; i++){
            for (int j = 0; j < col; j++){
                cin >> data[i][j];
            }
        }
    }

    void print() const{
        for (const auto& r: data){
            for (const auto& elem : r){
                cout << elem << " ";
            }
            cout << endl;
        }
    }

    friend void MatrixTranspose(const Matrix& matrix);
};

void MatrixTranspose(const Matrix& matrix){
    cout << "Original Matrix:" << endl;
    matrix.print();

    cout << "Transposed Matrix:" << endl;
    Matrix transpose(matrix.col, matrix.row);
    for (int i = 0; i < matrix.col; i++){
        for (int j = 0; j < matrix.row; j++){
            transpose.data[i][j] = matrix.data[j][i];
        }
    }
    transpose.print();
}

```

```
}
```

```
int main(){  
    int M, N;  
    cin >> M >> N;
```

```
    Matrix mat(M, N);  
    mat.setElements();
```

```
    MatrixTranspose(mat);
```

```
    return 0;
```

```
}
```

**Status :** Correct

**Marks :** 10/10