Date:

Assignment No.: 5

• Problem Statement:

Write a Program in C++ to create two different classes vector and matrix . The no. of rows and no. of columns can be taken as data members of both the classes . Two objects of both the classes should be created and then multiplication using friend function should be performed.

• Algorithm:

➔ **Name of the class:** Matrix, Vector
➔ **Private data members of the class:** col, vect[VECT_MAX]
➔ **Public member function of the class:** Vector()
➔ **Algorithm for constructor Vector():**

Step 1. col=101
Step 2. Input: “Enter size of the vector :”
Step 3. Read col
Step 4. Input: “ Enter elements : ”
Step 5. Repeat step6 for i = 0 to col
Step 6. Read vect[i]
End for lopp
Step 7. Call Friend Function : friend ostrem& operator<<( )
Step 8. Call Friend Function : friend Vector mult( )

➔ **Name of the class:** Matrix
➔ **Private data members of the class:**
col, row,mat[MAT_ROW][MAT_COL]
➔ **Public member function of the class:** Matrix()
➔ **Algorithm for constructor Matrix():**

Step 1. Input “Enter the dimention of the matrix(row,column): ”
Step 2. Read row,col
Step 3. Repeat through step 4 to step 5 for i = 0 to row
Step 4. Input “Enter row: ”
Step 5. Read mat[i][j]
End for loop
Step 6. Call Friend Function : friend Vector mult( )

➔ **Algorithm for method friend Vector mult( ):**

Step 1. Vector res = vector(0)
Step 2. If (v.col != m.row) then
Step 3. Print “ [Error] Dimention mismatch”
Step 4. Return res

Date:

End if
Step 5. Res.col = m.col
Step 6. Repeat through step 7 to step for i = 0 to res.col
Step 7. Res.vect[i]=0
Step 8. Repeat through step 9 to step for j = 0 to v.col
Step 9. Res.vect[i] += v.vect[j] * m.mat[j][i]
[End of inner for loop]
[End of outer for loop.]
Step 10. Return res.

➔ **Algorithm for method friend ostream& operator<<( ):**

Step 1. If (v.col ==0)then
Step 2. os =null vector
Step 3. Return os
End if
Step 4. Os= v.vect[0]
Step 5. Repeat through step 6 to step for i=1 to v.col
Step 6. Os =v.vect[i]
End for loop
Step 7. Return os

➔ **Algorithm for main function:**

Step 1. Call method Vector v()
Step 2. Call method Matrix m()
Step 3. Vector res=mult(v,m)
Step 4. Print “Result : res”

- **Source Code:**

```
#include <iostream>

using namespace std;

#define VECT_MAX 100
#define MAT_ROW 100
#define MAT_COL 100

class Matrix;

class Vector {
   private:
      int col;
      int vect[VECT_MAX];
   public:
      Vector(int c) {
```

Date:

```
        col = c;
    }

    Vector() {
        col = 101;
            cout << "Enter size of the vector : ";
            cin >> col;
            cout << endl;

        cout << "Enter " << col << " elements : ";

        for(int i = 0;i < col;i++) {
            cin >> vect[i];
        }

        cout << endl;
    }

    friend ostream& operator<<(ostream& os, Vector v);

    friend Vector mult(Vector v, Matrix m);
};

class Matrix {
    private:
        int col;
        int row;
        int mat[MAT_ROW][MAT_COL];
    public:
        Matrix() {
            cout << "Enter the dimension of the matrix (row & column) : ";
            cin >> row >> col;
            cout << endl;

            for(int i = 0;i < row;i++) {
                cout << "Enter row " << i << " : ";
                for(int j = 0;j < col;j++) {
                    cin >> mat[i][j];
                }
            }

        }

        friend Vector mult(Vector v, Matrix m);
};

ostream& operator<<(ostream& os, Vector v){
    if(v.col == 0) {
        os << "<null vector>";
        return os;
    }
```

Date:

```
    os << "[" << v.vect[0];
    for(int i = 1;i < v.col;i++) {
        os << ", " << v.vect[i];
    }
    os << "] ";
    return os;
}

Vector mult(Vector v, Matrix m) {
    Vector res = Vector(0);

    if(v.col != m.row) {
        cout << "[Error] Dimension mismatch!" << endl;
        return res;
    }

    res.col = m.col;

    for(int i = 0;i < res.col;i++) {
        res.vect[i] = 0;
        for(int j = 0;j < v.col;j++) {
            res.vect[i] += v.vect[j] * m.mat[j][i];
        }
    }

    return res;
}

int main() {
    Vector v;
    Matrix m;

    Vector res = mult(v, m);

    cout << "Result : " << res << endl;

    return 0;
}
```

- **Input & Output:**

Enter size of the vector : 4

Enter 4 elements : 21 56 32 12

Date:

Enter the dimension of the matrix (row & column) : 4 4

Enter row 0 : 14 15 20 21

Enter row 1 : 1 4 3 5

Enter row 2 : 16 24 10 9

Enter row 3 : 18 12 32 8

Result : [1078, 1451, 1292, 1105]

- **Discussion:**

1. In the above case, constructor are used. We also use the friends functions , and in this program we learn the uses of friend function.
2. The concepts of inheritance saves time and effort. It also enhances the reusability and reliability of the code.
3. When a class is inherited,constructor of the basr class is called first and the desturctor of the derived class is called first.