**Assignment no.:** 5

**Problem Statement:**

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 are created and then multiplication using friend function is to be perfomed.

**Algorithm:**

* Name of the class:- Matrix, Vector
* Private data members of the class:-

For Vector : col, vect[VECT\_MAX]

For Matrix : col, row,mat[MAT\_ROW][MAT\_COL]

* Public member function of the class:-

For Vector : Vector() -

For Matrix : Matrix() -

**Algorithm for constructor Vector():**

1. col=101
2. Input: “Enter size of the vector :”
3. Read col
4. Input: “ Enter elements : ”
5. Repeat step6 for i = 0 to col
6. Read vect[i]

End for lopp

1. Call Friend Function : friend ostrem& operator<<( )
2. Call Friend Function : friend Vector mult( )

**Algorithm for constructor Matrix():**

1. Input “Enter the dimention of the matrix(row,column): ”
2. Read row,col
3. Repeat through step 4 to step 5 for i = 0 to row
4. Input “Enter row: ”
5. Read mat[i][j]

End for loop

1. Call Friend Function : friend Vector mult( )

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

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

End if

1. Res.col = m.col
2. Repeat through step 7 to step for i = 0 to res.col
3. Res.vect[i]=0
4. Repeat through step 9 to step for j = 0 to v.col
5. 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<<( ):**

1. If (v.col ==0 )then
2. os =null vector
3. Return os

End if

1. Os= v.vect[0]
2. Repeat through step 6 to step for i=1 to v.col
3. Os =v.vect[i]

End for loop

1. Return os

**Algorithm for main function:**

1. Call method Vector v()
2. Call method Matrix m()
3. Vector res=mult(v,m)
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) {

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;

}

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 : 3

Enter 3 elements : 3

4

5

Enter the dimension of the matrix (row & column) : 3

3

Enter row 0 : 2

3

4

Enter row 1 : 5

6

7

Enter row 2 : 8

7

4

Result : [66, 68, 60]

**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 concepys 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.