

Computer Programming A, F
FAST-NU, Lahore, Spring 2018

Homework 3

class Matrix

Due Friday March 30 11:55 P.M.

Marked out of 50 points.

Create a class Matrix and provide all the necessary methods and operators required for the proper functioning of the following main. Matrix represents a mathematical matrix: an $m \times n$ two-dimensional array. The class uses an `int**mat`, and integers `m` and `n` as private members to store the data. The class includes a destructor and all necessary deep copy operators.

Where the matrices are being read from a file the following format is used: first line gives the value of `m` (number of rows), second line is the value of `n` (number of columns) and the matrix follows row by row in the next `m` lines, where each line contains `n` values separated by space.

```
4
3
5 9 2
3 8 1
2 1 33
5 4 12
```

```
int main()
{
    Matrix m1("mat1.txt"), m2("mat2.txt"), m3(2,5,0), m4;
    //here m3 2x5 matrix containing 0s
    if(m1!=m3){//assume compatibility for comparison
        m3=m3+m1; //assume compatibility for addition
    }

    cout<<"Result of addition is:"<<endl<<m3;

    m4=m3*m2; //assume compatibility for multiplication
    Matrix m5=(3*(m3++))+2;
    /* all entries of m3 are incremented by 1 but old values are used in computation since
    this is post increment; 3 is multiplied to each entry separately and then 2 is added to
    each entry of the resultant matrix which is then copied into m5*/
```

```

cout<<"Result of multiplication:"<<endl<<m4;
cout<<"Updated m3:"<<endl<<m3;
cout<<"Result of m5=(3*(m3++))+2:"<<endl<<m5;

//following code replaces all odd numbers in m4 by zeroes
for(int i=0;i<m5.rows();i++){
    for(int j=0;j<m5.cols();j++){
        if(m5[i][j]%2!=0){
            m5[i][j]=0;
        }
    }
}

cout<<"m5 after odds are zeroed:"<<endl<<m5;
Matrix m6 = Matrix::Identity(5,5);//a 5x5 identity matrix
cout<<"m6 is the identity matrix:"<<endl<<m6;
Matrix m7 = m6.transpose();
cout<<"m7 is the transpose of m6:"<<endl<<m7;
Matrix m8 = Matrix::Random(5,5,100);//a 5x5 matrix of random values below 100
cout<<"m8 is a 5x5 random matrix:"<<endl<<m8;
m7*=(++m8);
/*m7 is multiplied to m8 after each entry of m8 has been incremented by 1, the new
values of m8 are used in this computation since this is pre-increment. In this case, be
careful that the *=operator does not return Matrix by copy but const Matrix&*/
cout<<"Updated m8:"<<endl<<m8;
cout<<"Result of m7*=(++m8)"<<endl<<m7;

return 0;
}

```

THE END