College of Engineering

Department of Computer Science and Engineering

CMP220

Fall 2016

Jawahir Al Maazmi

@g00061542

Sec. 1

(array.h):

#ifndef ARRAY\_H

#define ARRAY\_H

#include<iostream>

using namespace std;

class Array2d

{

private:

int R = 3;

int C;

double \*arr[3];

public:

Array2d ();

Array2d(int c, double arrvalue);

Array2d(Array2d &x);

~Array2d();

Array2d &operator = (Array2d &x);

friend ostream &operator <<(ostream &out, Array2d &x);

};

#endif // !ARRAY\_H

(Array.cpp):

#include "Array.h"

Array2d::Array2d()

{

C = R;

for (int i = 0; i < R; i++)

{

arr[i] = new double[C];

for (int j = 0; j < C; j++)

{

arr[i][j] = 69;

}

}

}

Array2d::Array2d(int c, double arrvalue)

{

C = c;

for (int i = 0; i < R; i++)

{

arr[i] = new double[C];

for (int j = 0; j < C; j++)

{

arr[i][j] = arrvalue;

}

}

}

Array2d::Array2d(Array2d & x)

{

C = x.C;

R = x.R;

for (int i = 0; i < R; i++)

{

arr[i] = new double[C];

for (int j = 0; j < C; j++)

{

arr[i][j] = x.arr[i][j];

}

}

}

Array2d::~Array2d()

{

for (int i = 0; i < R; i++)

{

delete[]arr[i];

arr[i] = NULL;

}

}

Array2d & Array2d::operator=(Array2d & x)

{

for (int i = 0; i < x.R; i++)

{

delete[]arr[i];

arr[i] = NULL;

}

C = x.C;

R = x.R;

for (int i = 0; i < R; i++)

{

arr[i] = new double[C];

for (int j = 0; j < C; j++)

{

arr[i][j] = x.arr[i][j];

}

}

return \*this;

// TODO: insert return statement here

}

ostream & operator<<(ostream & out, Array2d & x)

{

for (int i = 0; i < x.R; i++)

{

out<<endl;

for (int j = 0; j < x.C; j++)

{

out << x.arr[i][j] << " ";

}

}

return (out);

// TODO: insert return statement here

}

(main.cpp):

#include "Array.h"

void main()

{

Array2d arrdef;

cout <<"Default Constructor Array: "<< arrdef << endl;

Array2d arrnon(4, 9.11);

cout << "Non-Default Constructor Array: " << arrnon << endl;

Array2d arrcpy(arrdef);

cout << "Copy Constructor Array (of the default constructor array): " << arrcpy << endl;

Array2d arrass = arrnon;

cout << "Assignment Operator Array (of the non-default constructor): " << arrass << endl;

}

