|  |  |  |
| --- | --- | --- |
| American University of SharjahSchool of Engineering Computer Science & Engineering Department  P. O. Box 26666, Sharjah, UAE |  | **LAB Instructor:** Eng. Sameer Alawnah **Office**: EB2-101  **Phone**: 971-6-515-2974  **e-mail**: salawnah@aus.edu  **Semester**: Fall 2016 |

**CMP 220 Intro. to Computer Science II**

**Quiz # 2**

**Jawahir Almaazmi**

**G00061542**

You have the following program skeleton:

#include <iostream>

using namespace std;

class Array2 { // Class declaration

friend ostream & operator <<(ostream & os, const Array2 & a);//print all elements in arr to the output stream os

friend istream & operator >>(istream & is, const Array2 & a);// read all elements of the array from the input stream is.

public:

Array2(int cols= 10); //Initialize the array with 0 values, default size =10

Array2(const Array2& c); //copy constructor

~Array2();//Destructor

int getCols() const{

return cols;

}// return the size of the array.

int getAt(int i,int j)

{

if (i<0 || i>3)

{

cout << "Index out of range error!\n";

exit(-1);

}

if (j<0 || j>cols)

{

cout << "Index out of range error!\n";

exit(-1);

}

return arr[i][j];

}

void setAt(int i,int j, int value)

{

if (i<0 || i>3)

{

cout << "Index out of range error!\n";

exit(-1);

}

if (j<0 || j>cols)

{

cout << "Index out of range error!\n";

exit(-1);

}

arr[i][j] = value;

}

private:

int cols; // size of the created array

int \* arr[3];

};

Array2 increment(Array2 a)

{

for (int i = 0; i<3; i++){

for (int j = 0; j < a.getCols();j++)

a.setAt(i,j, a.getAt(i,j) + 1);

}

return a;

}

istream & operator >>(istream & is, const Array2 & a){

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < a.getCols(); j++)

{

is >> a.arr[i][j];

}

}

return is;

}

ostream & operator <<(ostream & os, const Array2 & a)

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < a.getCols(); j++)

{

os << a.arr[i][j] << " ";

}

os << endl;

}

return os;

}

int main()

{

Array2 a(2);

cout << "Please enter A:\n";

cin >> a;

cout << "A = \n" << a << endl;

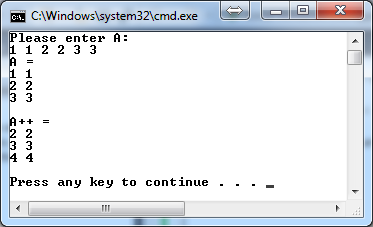
Array2 b = increment(a);

cout << "A++ \n= " << b << endl;

return 0;

}

1. Implement all unimplemented functions/operators
2. Run the program and make sure to give you the same output below without any runtime/ compilation errors:



Note: you CAN change the class definition and implementation but you are NOT allowed to modify the increment and the main functions, changing them will result in ZERO grade.

Good Luck ☺

#include <iostream>

using namespace std;

class Array2 { // Class declaration

friend ostream & operator <<(ostream & os, const Array2 & a);//print all elements in arr to the output stream os

friend istream & operator >>(istream & is, Array2 & a);// read all elements of the array from the input stream is.

public:

Array2(int cols = 10) //Initialize the array with 0 values, default size =10

{

C = cols;

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

{

arr[i] = new int[C];

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

{

arr[i][j] = 0;

}

}

}

Array2(int c, int value)// non-default constructor

{

cols = c;

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

{

arr[i] = new int[c];

for (int j = 0; j < cols; i++)

{

arr[i][j] = value;

}

}

}

Array2(const Array2& c)//copy constructor

{

cols = c.cols;

rows = c.rows;

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

{

arr[i] = new int[cols];

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

{

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

}

}

}

~Array2()//Destructor

{

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

{

delete[]arr[i];

arr[i] = NULL;

}

}

int getCols() const{

return cols;

}// return the size of the array.

int getAt(int i, int j)

{

if (i<0 || i>3)

{

cout << "Index out of range error!\n";

exit(-1);

}

if (j<0 || j>cols)

{

cout << "Index out of range error!\n";

exit(-1);

}

return arr[i][j];

}

void setAt(int i, int j, int value)

{

if (i<0 || i>3)

{

cout << "Index out of range error!\n";

exit(-1);

}

if (j<0 || j>cols)

{

cout << "Index out of range error!\n";

exit(-1);

}

arr[i][j] = value;

}

private:

int cols; // size of the created array

int \* arr[3];

int rows=3;

int C;

};

Array2 increment(Array2 a)

{

for (int i = 0; i<3; i++){

for (int j = 0; j < a.getCols(); j++)

a.setAt(i, j, a.getAt(i, j) + 1);

}

return a;

}

istream & operator >>(istream & is, Array2 & a){

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < a.getCols(); j++)

{

is >> a.arr[i][j];

}

}

return is;

}

ostream & operator <<(ostream & os, const Array2 & a)

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < a.getCols(); j++)

{

os << a.arr[i][j] << " ";

}

os << endl;

}

return os;

}

int main()

{

Array2 a(2);

cout << "Please enter A:\n";

cin >> a;

cout << "A = \n" << a << endl;

Array2 b = increment(a);

cout << "A++ \n= " << b << endl;

return 0;

}

ANSWER: fml

#include <iostream>

using namespace std;

class Array2 { // Class declaration

friend ostream & operator <<(ostream & os, const Array2 & a);//print all elements in arr to the output stream os

friend istream & operator >>(istream & is, Array2 & a);// read all elements of the array from the input stream is.

public:

Array2(int c= 10) //Initialize the array with 0 values, default size =10

{

cols = c;

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

{

arr[i] = new int[c];

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

{

arr[i][j] = 0;

}

}

}

Array2(int c, int value)// non-default constructor

{

cols = c;

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

{

arr[i] = new int[c];

for (int j = 0; j < cols; i++)

{

arr[i][j] = value;

}

}

}

Array2(const Array2& c)//copy constructor

{

cols = c.cols;

rows = c.rows;

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

{

arr[i] = new int[cols];

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

{

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

}

}

}

~Array2()//Destructor

{

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

{

delete[]arr[i];

arr[i] = NULL;

}

}

int getCols() const{

return cols;

}// return the size of the array.

int getAt(int i, int j)

{

if (i<0 || i>3)

{

cout << "Index out of range error!\n";

exit(-1);

}

if (j<0 || j>cols)

{

cout << "Index out of range error!\n";

exit(-1);

}

return arr[i][j];

}

void setAt(int i, int j, int value)

{

if (i<0 || i>3)

{

cout << "Index out of range error!\n";

exit(-1);

}

if (j<0 || j>cols)

{

cout << "Index out of range error!\n";

exit(-1);

}

arr[i][j] = value;

}

private:

int cols; // size of the created array

int \* arr[3];

int rows=3;

};

Array2 increment(Array2 a)

{

for (int i = 0; i<3; i++){

for (int j = 0; j < a.getCols(); j++)

a.setAt(i, j, a.getAt(i, j) + 1);

}

return a;

}

istream & operator >>(istream & is, Array2 & a){

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < a.getCols(); j++)

{

is >> a.arr[i][j];

}

}

return is;

}

ostream & operator <<(ostream & os, const Array2 & a)

{

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < a.getCols(); j++)

{

os << a.arr[i][j] << " ";

}

os << endl;

}

return os;

}

int main()

{

Array2 a(2);

cout << "Please enter A:\n";

cin >> a;

cout << "A = \n" << a << endl;

Array2 b = increment(a);

cout << "A++ \n= " << b << endl;

return 0;

}

