|  |  |  |
| --- | --- | --- |
| American University of SharjahSchool of Engineering Computer Engineering Department  P. O. Box 26666 Sharjah, UAE |  | **Lab Instructor:** Eng. Sameer Alawnah  **Office**: EB2 - 101  **Phone**: 06-515 2974  **e-mail**: salawnah@aus.edu  **Semester**: Fall 2016 |

**CMP 220L – Introduction to Computer Science II**

**Operator overloading and Dynamic memory**

**LAB Assignment #9**

**Exercise 1**

Implement the **Array** class that is defined below and test it in the main program. The main program must test all the member functions including the overloaded operators that are defined in the class.

#include <iostream>

using namespace std;

class Array { // Class declaration

friend const Array operator+(const Array & a1, const Array &a2);

friend const Array operator-(const Array &a1,const Array &a2);

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

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

friend bool operator==(const Array & a, const Array & b) ;//Equality test, return true if all elements in a.arr are matching the corresponding elements in the array b.arr. if the a.size is not equal to b.size then you have to reutrn false.

friend bool operator!=(const Array & a, const Array & b) ;//Not equal test, return true if a.size not equal to b.size or at least one element in a.arr not matching the corrsponding elements in b.arr

friend const Array operator-(const Array & a);//Negate (unary operation), create an array that contains a negated elements of a.arr and reutrn it

public:

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

Array(const Array & a); // copy constructor

~Array();//Destructor

int getSize();// return the size of the array.

void operator=(const Array & a);// Assignment operator, if the size of the array a is greater than size, then you have to delete the current array and allocate memory that can fit the content of the array a.

Array operator++();//pre increment, increment all elements in the arr

void operator+=(const Array & a);// add each element in a.arr to the corresponding element in arr

private:

int size; // size of created array

int \* arr;

};

Test your class with the following main():

int main()

{

Array a(5), b(5);

cout << "Please enter the content of the first array:\n";

cin >> a;

cout << "Please enter the content of the second array:\n";

cin >> b;

cout << "A = " << a << " B = " << b << endl;

Array c = a + b;

cout << "C = A + B = " << c << endl;

++a;

cout << "++A\n";

cout << "A after ++A = " << a << endl;

Array d = (a-b)+c;

cout<<" D = (a-b) + c = "<<d<<endl;

cout << "-C = " << -c << endl;

cout << "C = " << c << endl;

c = a;

cout << "C = A = " << c << endl;

cout << "C == A = " << (c == a) << endl;

cout << "C != A = " << (c != a) << endl;

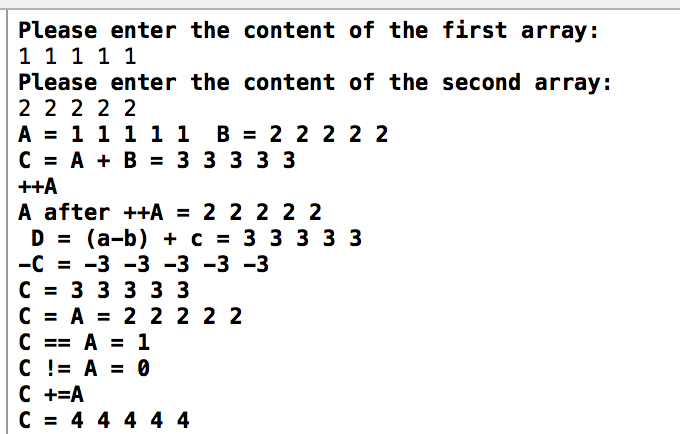
c += a;

cout << "C +=A\n";

cout << "C = " << c << endl;

}

you should get an output like the following:



Good Luck ☺

Arr.h:

#ifndef ARR\_H

#define ARR\_H

#include <iostream>

using namespace std;

class Array { // Class declaration

friend const Array operator+(const Array & a1, const Array &a2);

friend const Array operator-(const Array &a1, const Array &a2);

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

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

friend bool operator==(const Array & a, const Array & b);//Equality test, return true if all elements in a.arr are matching the corresponding elements in the array b.arr. if the a.size is not equal to b.size then you have to reutrn false.

friend bool operator!=(const Array & a, const Array & b);//Not equal test, return true if a.size not equal to b.size or at least one element in a.arr not matching the corrsponding elements in b.arr

friend const Array operator-(const Array & a);//Negate (unary operation), create an array that contains a negated elements of a.arr and reutrn it

public:

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

Array(const Array & a); // copy constructor

~Array();//Destructor

int getSize(const int size);// return the size of the array.

void operator=(const Array & a);// Assignment operator, if the size of the array a is greater than size, then you have to delete the current array and allocate memory that can fit the content of the array a.

Array operator++()//pre increment, increment all elements in the arr

{

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

{

arr[i] = ++arr[i];

}

return (size);

}

void operator+=(const Array & a);// add each element in a.arr to the corresponding element in arr

private:

int size; // size of created array

int \*arr;

};

#endif

Arr.cpp:

#include "arr.h"

Array::Array(int Size)

{

size = Size;

arr = new int[size];

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

{

arr[i] = 0;

}

}

Array::Array(const Array & a) // copy constructor

{

size = a.size;

arr = new int[size];

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

{

arr[i] = a.arr[i];

}

}

Array::~Array()//Destructor

{

delete[]arr;

arr = NULL;

}

int getSize(const int size)// return the size of the array.

{

return size;

}

void Array::operator=(const Array & a)

{

if (a.size != size )

{

size = a.size;

delete[]arr;

arr = new int[size];

}

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

{

arr[i] = a.arr[i];

}

}

void Array::operator+=(const Array & a)// add each element in a.arr to the corresponding element in arr

{

for (int i = 0; i < a.size; i++)

{

arr[i] = a.arr[i] + arr[i];

}

}

const Array operator+(const Array & a1, const Array &a2)

{

if (a1.size == a2.size)

{

Array result(a1.size);

for (int i = 0; i < result.size; i++)

{

result.arr[i] = a1.arr[i] + a2.arr[i];

}

return (result);

}

else

{

exit(1);

}

}

const Array operator-(const Array &a1, const Array &a2)

{

if (a1.size == a2.size)

{

Array result(a1.size);

for (int i = 0; i < result.size; i++)

{

result.arr[i] = a1.arr[i] - a2.arr[i];

}

return (result);

}

else {

exit(1);

}

}

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

{

int size;

size = a.size;

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

{

os << a.arr[i];

}

return (os);

}

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

{

int size;

size = a.size;

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

{

is >> a.arr[i];

}

return (is);

}

bool operator==(const Array & a, const Array & b)//Equality test, return true if all elements in a.arr are matching the corresponding elements in the array b.arr. if the a.size is not equal to b.size then you have to reutrn false.

{

if (a.size == b.size)

{

return true;

}

else

{

return false;

}

}

bool operator!=(const Array & a, const Array & b)//Not equal test, return true if a.size not equal to b.size or at least one element in a.arr not matching the corrsponding elements in b.arr

{

if (a.size != b.size)

{

return true;

}

else

{

return false;

}

}

const Array operator-(const Array & a)//Negate (unary operation), create an array that contains a negated elements of a.arr and reutrn it

{

Array res(a);

for (int i = 0; i < res.size; i++)

{

res.arr[i] = -res.arr[i];

}

return (res);

}

Main.cpp

#include "arr.h"

int main()

{

Array a(5), b(5);

cout << "Please enter the content of the first array:\n";

cin >> a;

cout << "Please enter the content of the second array:\n";

cin >> b;

cout << "A = " << a << " B = " << b << endl;

Array c = a + b;

cout << "C = A + B = " << c << endl;

++a;

cout << "++A\n";

cout << "A after ++A = " << a << endl;

Array d = (a - b) + c;

cout << " D = (a-b) + c = " << d << endl;

cout << "-C = " << -c << endl;

cout << "C = " << c << endl;

c = a;

cout << "C = A = " << c << endl;

cout << "C == A = " << (c == a) << endl;

cout << "C != A = " << (c != a) << endl;

c += a;

cout << "C +=A\n";

cout << "C = " << c << endl;

}

