Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 5, варіант 23

З дисципліни « ПРОГРАМУВАННЯ, ЧАСТИНА 2 (ОБ’ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ) »

На тему: «Перевантаження операторів»

Виконав: ст. гр. КІ-15

Пилипів Р. С.

Перевірив: викладач

Козак Н.Б.

Львів – 2020

**Завдання:**

23. +=, %, ==, !

**Код програми:**

//Array.h

#include<iostream>

#include<stdlib.h>

#include<time.h>

using namespace std;

class Array

{

private:

int\* ptr;

int size;

public:

Array(int s = 10);

Array(Array& arr);

virtual ~Array();

friend ostream& operator<< (ostream& output, Array& arr);

void Randomize(int num = 10);

Array& operator= (Array& arr);

Array& operator+= (Array& arr);

Array operator% (Array& arr);

Array& operator== (Array& arr);

Array operator!();

};

//Array.cpp

#include "Array.h"

Array::Array(int s)

{

size = s;

ptr = new int[size];

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

{

ptr[i] = 0;

}

cout << "Constructor" << endl;

}

//----------------------------------------------------------------------

Array::Array(Array& arr)

{

size = arr.size;

ptr = new int[size];

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

{

ptr[i] = arr.ptr[i];

}

cout << "Copy Constructor" << endl;

}

//----------------------------------------------------------------------

Array::~Array()

{

delete[] ptr;

cout << "Destructor" << endl;

}

//----------------------------------------------------------------------

void Array::Randomize(int num)

{

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

{

ptr[i] = rand() % num;

}

}

//----------------------------------------------------------------------

ostream & operator<< (ostream & output, Array & arr)

{

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

{

output << arr.ptr[i] << " ";

}

output << endl;

return output;

}

//----------------------------------------------------------------------

Array& Array::operator= (Array& arr)

{

if (this != &arr)

{

delete[] ptr;

size = arr.size;

ptr = new int[size];

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

{

ptr[i] = arr.ptr[i];

}

}

return \*this;

}

//----------------------------------------------------------------------

Array& Array::operator+= (Array& arr)

{

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

{

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

}

cout << "Operation += : " << endl;

return \*this;

}

//----------------------------------------------------------------------

Array Array::operator%(Array& arr)

{

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

{

if (arr.ptr[i] != 0)

{

ptr[i] = ptr[i] % arr.ptr[i];

}

else ptr[i] = 0;

}

cout << "Operation % :" << endl;

return \*this;

}

//----------------------------------------------------------------------

Array& Array::operator==(Array& arr)

{

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

{

ptr[i] = arr.ptr[i];

}

cout << "Operation == :" << endl;

return \*this;

}

//----------------------------------------------------------------------

Array Array::operator!()

{

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

{

ptr[i] = !ptr[i];

}

cout << "Operation ! : " << endl;

return \*this;

}

//main.cpp

#include "Array.h"

int main(void)

{

Array a(5);

a.Randomize(5);

Array b(5);

b.Randomize(6);

Array c(5);

Array d(5);

cout << "Array a: " << endl << a;

cout << "Array b: " << endl << b;

c = a;

d = b;

d.operator+=(c);

cout << d;

c = a;

d = b;

d.operator%(c);

cout << d;

c = a;

d = b;

d.operator==(c);

cout << d;

c.Randomize(2);

cout << "Array c: " << endl << c;

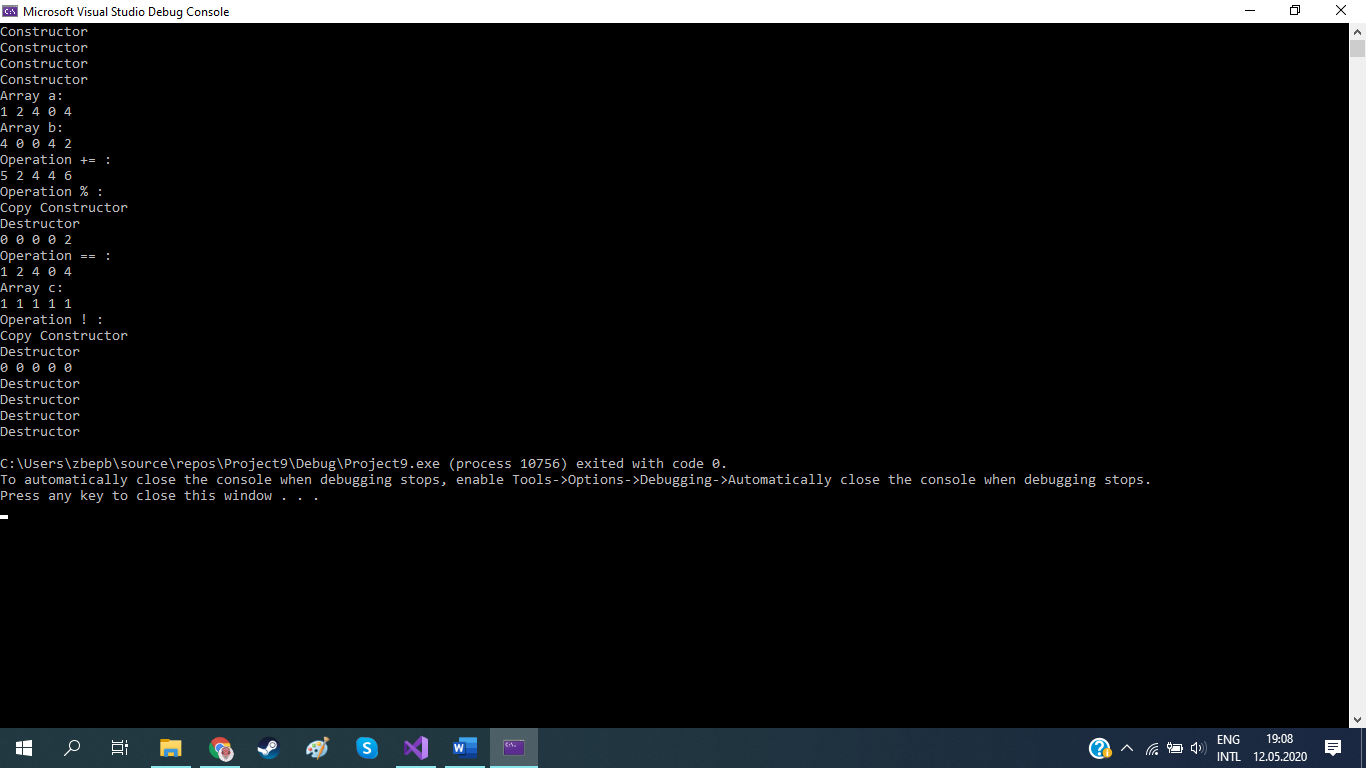
c.operator!();

cout << c;

return 0;

}

**Вікно результату:**



**Висновок.** На цій лабораторній роботі я познайомився з перенавантаженням операторів.