Laboratory Work No. 8

Operator Overloading

This laboratory work covers the following concepts in C++ programming language:

- class declaration (access specifiers: public, private)
- data members, member functions, friend functions, constructors, destructor
- operator overloading

⇨ Create a Win32 Console application and an empty C++ source file in Visual Studio IDE to be able to start typing programs.

Task-1: Consider below partly given class declaration.

```
#define MAX_SIZE 100
class Array {
        friend … operator<< (…);       //overload “stream insertion” operator
        friend … operator>> (…);       //overload “stream extraction” operator
    public:
        Array();                       //default constructor
        Array(…);                      //parameterized constructor
        Array(…);                      //copy constructor
        ~ Array();                     //destructor
        … getSize();                   //a constant member function
        … setSize(…);                  //a non-constant member function
        … operator==(…);               //overload “is equal” operator
        … operator!=(…);               //overload “is not equal” operator
        … operator[](…);               //overload “subscript” operator as a non-constant l-value
        … operator[](…);               //overload “subscript” operator as a constant r-value
        … operator()(…);               //overload “parenthesis” operator (passing index and value to be stored)
        … operator++(…);               //overload “pre-increment” operator
        … operator++(…);               //overload “post-increment” operator
    private:
        int list[MAX_SIZE];
        int size;
};
```

a. Write definitions of the member functions listed above.
b. Rewrite the program by separating the implementation file from the interface using a header file.

⇨ Provide a driver program to test each implementation.

[Optional task] Reconsider the Array class declaration. Convert the implementation into a dynamic array form!