

Lab 2. Simple classes. Methods.

Write a class in C++ that has the following definition:

```
class NumberList
{
    int numbers[10];
    int count;
public:
    void Init();           // count will be 0
    bool Add(int x);       // adds X to the numbers list and increase the data
                           // member count.
                           // if count is bigger or equal to 10, the function will return
false
    void Sort();           // will sort the numbers vector
    void Print();          // will print the current vector
}
```

Organize the code in the following way:

- a header file called **NumberList.h**
- a cpp file called **NumberList.cpp** that contains the source code for class **NumberList**
- a main file called **main.cpp** that contains the main function and has an example on how to use **NumberList**. The example must include using all methods from the class.
- make sure that precompile headers are NOT used (for Visual Studio setup).

Design a C++ class (following the previous problem) that reflects the properties of a student. The class should include the following:

- methods to set and get the name of the student
- methods to set and get the grade for mathematics (a grade must be a float value from 1 to 10)
- methods to set and get the grade for English (a grade must be a float value from 1 to 10)
- methods to set and get the grade for History (a grade must be a float value from 1 to 10)
- a method that retrieves the average grade
- 5 global functions that compares two students in terms of name, grades, average. If two students are equal the return value of such a function will be 0. If the first student is bigger than the second one, the return value will be 1, otherwise -1.

Make sure that you have the following:

- a header file for the class
- a cpp file for the methods specific to the class
- a header file for the global functions
- a cpp file for the global functions implementation
- a main.cpp file that shows how the methods and global functions can be used.