

**Sabanci University**  
Faculty of Engineering and Natural Sciences  
CS204 Advanced Programming  
Fall 2019-2020

Homework 4 – Constructors and  
Operator Overloading  
Due: **30/10/2019, 11:55pm**

**PLEASE NOTE:**

**Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!**

**You HAVE TO write down the code on your own.**

**You CANNOT HELP any friend while coding.**

**Plagiarism will not be tolerated!**

**Introduction**

This homework's aim is to make you familiar with the fundamentals of Object-Oriented Programming: **constructors**, **destructors** and **operator overloading**. You will be given a sorted linked list class which you will be asked to create constructor, destructor and overload some operators.

Along with the PDF document, you are given a header file **SortedArray.h** and a corresponding cpp file **SortedArray.cpp**. Also another cpp file is supplied to you named **Test.cpp** for testing your code. You will be editing SortedArray.h and SortedArray.cpp files. You won't be editing the Test.cpp file. Also output of the Test.cpp will be given so that you can check your code works correctly.

You need to implement **(deep) copy constructor** and **destructor** for SortedArray class. There are some comments in the given header and cpp files, use them as guidelines. You will also need to overload **assignment** (=) and **plus** (+) operators. After you implement copy constructor, destructor, assignment and addition operators, you should be able to do the following example operations.

```
SortedArray a2, a3, a4;  
SortedArray a1(a2); // copy constructor  
a4 = a1 + a2 + a3; // sorted arrays will be merged and result will also be sorted  
a1 = a1 + 10; // new integer value (in this case 10) will be inserted to sorted array  
a2 = 5 + a1; // new integer value (in this case 5) will be inserted to sorted array  
a1 = a2 = a3; //cascading assignment
```

**Output**

Output of the Test.cpp is given in the following image, which is also provided in the output.txt file.

```
Array1: 2 4 5 7 10

Array1 = Array1 + 20
Array1: 2 4 5 7 10 20

Array1 = 6 + Array1
Array1: 2 4 5 6 7 10 20

Array2: 0 2 8 9 10

Array1 + Array2 = 0 2 4 5 6 7 8 9 10 20

Array3: 20 30 40 50 70

Array1 + Array2 + Array3 = 0 2 4 5 6 7 8 9 10 20 30 40 50 70

1 + Array1 + 56 = 1 2 4 5 6 7 10 20 56

After => Array1 = Array2 = Array3
Array1: 20 30 40 50 70
Array2: 20 30 40 50 70
Array3: 20 30 40 50 70

Array4 is generated with copy constructor of Array3
Array4: 20 30 40 50 70

Press any key to continue . . .
```

### Some Important Rules

In order to get a full credit, your programs must be efficient and well presented, presence of any redundant computation or bad indentation, or missing, irrelevant comments are going to decrease your grades. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate.

When we grade your homeworks we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we are going to run your programs in *Release* mode and **we may test your programs with very large test cases**.

### What and where to submit (PLEASE READ, IMPORTANT)

You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework. It'd be a good idea to write your name and last name in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do

not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.  
Name your cpp file that contains your program as follows:

***"SUCourseUserName\_YourLastname\_YourName\_HWnumber.cpp"***

Your SUCourse user name is actually your SUNet username that is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is valent, name is Valentina, and last name is Tereşkova, then the file name must be:

***Valent\_Tereskova\_Valentina\_hw1.cpp***

Do not add any other character or phrase to the file name. Make sure that this file is the latest version of your homework program. Compress this cpp file using WINZIP or WINRAR programs. Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension. Check that your compressed file opens up correctly and it contains your cpp file.

You will receive no credits if your compressed zip file does not expand or it does not contain the correct file. The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows:

***SUCourseUserName\_YourLastname\_YourName\_HWnumber.zip***

For example zubzipler\_Zipleroglu\_Zubeyir\_hw1.zip is a valid name, but

***hw1\_hoz\_HasanOz.zip, HasanOzHoz.zip***

are **NOT** valid names.

**Submit via SUCourse ONLY!** You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!

CS204 Team (M. Yusa Erguven, Kamer Kaya)