**Sabancı University**

Faculty of Engineering and Natural Sciences

**CS204 Advanced Programming**

**Fall 2018-2019**

**Homework 8 – Inheritance & Polymorphism**

**Due: 25/12/2018 - 23:55**

(One day late submission penalty: -10%)

|  |
| --- |
| **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!** |

1. **Introduction**

The aim of this homework is to practice inheritance and polymorphism. The goal is to implement a base abstract class with three subclasses. The base class will be a container class which is an abstract class. Then you will implement a **sorted LinkedList, Stack** and **Queue** classes that inherit from the **Container** class. The main function is given and you are only supposed to implement the classes.

1. **Program Flow**

The program will iteratively ask the user to select a data structure and then an operation. The operations are:

**Insert:** Insert an integer to the corresponding data structure. You can assume that no duplicates will be inserted.

**Delete:** Delete an integer from the corresponding data structure.

**Print:** Print the corresponding data structure.

The base class will define these functions as pure virtual functions.Then all the subclasses will implement these functions according to their definitions. Also, every data structure should prompt a message explaining the operation being carried out. For further information, refer to the sample runs.

1. **Sample Runs**

Sample 1

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 10**

**10 is inserted in the linked list**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 5**

**5 is inserted in the linked list**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 19**

**19 is inserted in the linked list**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 7**

**7 is inserted in the linked list**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **2**

**Printing linked list**

**5 7 10 19**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **1**

Enter the number to be deleted: **10**

**10 is deleted from the linkedlist**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **1**

Enter the number to be deleted: **7**

**7 is deleted from the linkedlist**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **0**

Enter the operation(0-insert, 1-delete, 2-print): **2**

**Printing linked list**

**5 19**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **3**

**Destructing linkedlist**

**Destructing stack**

**Destructing queue.**

**Exiting..**

Sample 2

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 10**

**Inserting 10 to stack**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 5**

**Inserting 5 to stack**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 19**

**Inserting 19 to stack**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 7**

**Inserting 7 to stack**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **2**

**Printing stack**

**7 19 5 10**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **1**

**Enter the number to be deleted: 19**

**19 is deleted from the stack**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **1**

**Enter the number to be deleted: 10**

**10 is deleted from the stack**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **1**

Enter the operation(0-insert, 1-delete, 2-print): **2**

**Printing stack**

**7 5**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **3**

**Destructing linkedlist**

**Destructing stack**

**Destructing queue.**

**Exiting..**

Sample 3

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 10**

**Inserting 10 to queue**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 5**

**Inserting 5 to queue**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 19**

**Inserting 19 to queue**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **0**

**Enter the number to be inserted: 7**

**Inserting 7 to queue**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **2**

**Printing queue**

**10 5 19 7**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **1**

**Enter the number to be deleted: 5**

**5 is deleted from the queue.**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **1**

**Enter the number to be deleted: 19**

**19 is deleted from the queue.**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **2**

Enter the operation(0-insert, 1-delete, 2-print): **2**

**Printing queue**

**10 7**

Enter your selection(0-LinkedList, 1-Stack, 2-Queue, 3-Exit): **3**

**Destructing linkedlist**

**Destructing stack**

**Destructing queue.**

**Exiting..**

**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 homework, we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we may 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 cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

***Cago\_Ozbugsizkodyazaroglu\_Caglayan\_hw2.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

***Hw2\_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 (Mustafa Kemal Taş, Kamer Kaya)