Sabancı University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Fall 2016-2017

Homework 4–Operator and Iterator class

Due: 7/11/2016, 23:30
(Late submission penalty: -20%)

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: The aim of this homework is to get familiar with operators and Iterator class. In this homework, you are asked to implement some overloading operators (union, difference, intersection, and [] index search) as a member function or free function. You need to use an Iterator to implement a function for the last operator ([]). The details will be explained in subsequent sections of this homework.

The Data Structures: In this homework, you must represent the students who registered for a course as a linked list (regular one-way linked list). As a result, you are going to implement one node type that stores students' information with its ID only. The IDs of students in each linked list must be ***ordered and unique*** (means NO duplication must be exist in a list). You are not allowed to use arrays, vectors, and similar containers. The operators to be implemented will be explained in next section. The structure of a node and header of the linked list class exist in ***linkedlist.h*** file to give you a clue. You are free to add any member function, friend function or class to the header file.

The Program Flow: You have to generate two linked lists for ***two courses***; each list consists of students who register for the corresponding course. Each node of the linked list contains only the ID of the registered student.

At the very beginning of your program, the user will be asked to enter a course name (the name of the course has no effect in the flow of the program; it is only asked for clarification). Afterwards, your program will ask for students who registered for the course.The number of

students to be asked for the course depends on a user. As long as a user has not entered the "^z", the program keep getting student ID(use string type for IDs).
We have two lists list1, and list2 for optional two courses. The following is an example showing list of registered students for courses CS201 and CS204 respectively. There is no need to check validity of inputs, i.e., you don't need to check the characters, numbers etc. But you need to keep your list sorted after each insertion.

Enter the name of the course: **CS201**
Course CS201 registration list
Please enter a studentID !
12
Please enter a student ID!
45
Please enter a student ID!
78
Please enter a student ID!
85
Please enter a student ID!
^Z
List of students who registered for course CS201 is:
12
45
78
85

Enter the name of the course: **CS204**
Course CS204 registration list
Please enter a studentID !
21
Please enter a student ID!
43
Please enter a student ID!
12
Please enter a student ID!
85
Please enter a student ID!
^Z
List of students who registered for course CS204 is:
12
21
43
85

Table 1

After two linked lists are generated, a ***menu*** containing ***5*** options will be displayed. The first four options are related to operators which are applied on the linked lists. After each menu option is selected and the required processing is performed, the menu should be displayed and a new option will be selected continuously until the use enters 5 to Exit. In case the user enters a text rather than menu options, an appropriate message must be displayed and return to menu again. All the samples for these operators are generated using two lists mentioned in Table 1. Details about each menu option are as follow:

1. **+ (union, member function)** It applies on two lists and extract those students IDs which are exist in list1***OR*** list2 (list1+list2). The result must be added to a new linked list (no duplication meaning that students who register for both courses must be occurred once the list) in sorted order and printed on the screen.

```
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 1
Union of two lists are:
12
21
43
45
78
85
```

Table 2

2. **– (difference, member function)** It applies on two linked lists and extract all students IDs which are exist in ***lhs*** but not in ***rhs*** (list1-list2 return all IDs which are exist in list1 but not list2). The order of **lhs** and **rhs** are important while taking difference of two sets since (list1-list2) is not equal to (list2-list1). The result must be added to a new linked list in sorted order and printed on the screen. When this option is selected, the list which is going to be the***lhs*** must be taken from user (1 for the first list and 2 for the second list).

```
*  1. Union(+)           *
*  2. Difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *

Your choice: 2
Enter the lhs list: 1
difference of list1-list2 is:
```

```
45
78



*****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
*****************************

Your choice: 2
Enter the lhs list: 2
difference of list2-list1 is:
21
43
```

Table 3

3. **& (intersection, free function)** It applies on two linked lists and extracts students IDs which are exist in both lists (students who are registered for both courses). The result must be added to a new linked list in sorted order and printed on the screen.

```
*****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
*****************************

Your choice: 3
Intersection of list1 and list2 is:
12
85
```

4. **[] (index search**, **member function, with an iterator**) Overload this operator as a member function. This operator is applied on a list and returns the ID of the student at the ***index*** position of the list. As an example list1[2] returns the ID of student at the third (index starts from 0) position of the list1. If such index does not exist in the list (means the list does not have these many nodes), it must show an appropriate message. When option 4 is selected, two inputs must be given by user. The first is a list which [] operator is applied on, and the second is the index. **You need to use an Iterator to implement this function.** The result must be printed on the screen.

```
*****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
```

```
*  4. [] operator        *
*  5. Exit              *
****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 0
The index 0 of the list is:
12
****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
****************************
Your choice: 4
Enter the list number: 1
Enter an index number: 3
The index 3 of the list is:
85
```

5. Exit
   When this option is selected, your program is terminated. In order to make sure that you make no memory leak, your program must return all the dynamically allocated memory to heap before the termination.

```
****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
****************************

Your choice: 5
Program Exiting...
Press any key to continue . . .
```

Sample Runs

Sample run 1

```
Enter the name of the course: CS201
Course CS201 registration list
Please enter a student ID !
^Z
List of students who registered for course CS201 is:
List is empty
```

```
Enter the name of the course: CS204
Course CS204 registration list
Please enter a student ID !
^Z
List of students who registered for course CS204 is:
List is empty

****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 1
Union of two lists are:
List is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 2
Enter the lhs list: 1

difference of list1-list2 is:
List is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 3
Intersection of list1 and list2 is:
List is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 0
```

```
The index 0 of the list is:
The list is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 2
Enter an index number: 0
The index 0 of the list is:
The list is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice:
```

Sample Run 2

```
Enter the name of the course: cs405
Course cs405 registration list
Please enter a student ID !
^Z
List of students who registered for course
List is empty

Enter the name of the course: IE501
Course IE501 registration list
Please enter a student ID !
12
Please enter a student ID!
14
Please enter a student ID!
35
Please enter a student ID!
24
Please enter a student ID!
78
Please enter a student ID!
57
Please enter a student ID!
^Z
List of students who registered for course
12
14
24
35
```

```
57
78

****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 1
Union of two lists are:
12
14
24
35
57
78
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 2
Enter the lhs list: 1

difference of list1-list2 is:
List is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 2
Enter the lhs list: 2

difference of list2-list1 is:
12
14
24
35
57
78
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
```

```
*  5. Exit              *
****************************

Your choice: 3
Intersection of list1 and list2 is:
List is empty
****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 3
The index 3 of the list is:
The list is empty
****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
****************************

Your choice: 4
Enter the list number: 2
Enter an index number: 5
The index 5 of the list is:
78
****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
****************************

Your choice: 4
Enter the list number: 2
Enter an index number: 7
The index 7 of the list is:
 out of bound
****************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
*  4. [] operator       *
*  5. Exit              *
****************************
Your choice: 5
Program Exiting...
Press any key to continue . . .
```

Sample run 3

```
Enter the name of the course: CS201
Course CS201 registration list
Please enter a student ID !
12
Please enter a student ID!
45
Please enter a student ID!
78
Please enter a student ID!
85
Please enter a student ID!
24
Please enter a student ID!
^Z
List of students who registered for course CS201 is:
12
24
45
78
85

Enter the name of the course: CS204
Course CS204 registration list
Please enter a student ID !
12
Please enter a student ID!
85
Please enter a student ID!
96
Please enter a student ID!
35
Please enter a student ID!
74
Please enter a student ID!
21
Please enter a student ID!
10
Please enter a student ID!
^Z
List of students who registered for course CS204 is:
10
12
21
35
74
85
96

**************************
*  1. union(+)          *
*  2. difference(-)     *
*  3. Intersection (&)  *
```

```
*  4. [] operator        *
*  5. Exit              *
****************************

Your choice: 1
Union of two lists are:
10
12
21
24
35
45
74
78
85
96
****************************
*  1. union(+)          *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit              *
****************************

Your choice: 2
Enter the lhs list: 1

difference of list1-list2 is:
24
45
78
****************************
*  1. union(+)          *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit              *
****************************

Your choice: 2
Enter the lhs list: 2

difference of list2-list1 is:
10
21
35
74
96
****************************
*  1. union(+)          *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit              *
****************************
```

```
Your choice: 3
Intersection of list1 and list2 is:
12
85
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 5
The index 5 of the list is:
 out of bound
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 4
The index 4 of the list is:
85
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 0
The index 0 of the list is:
12
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 2
Enter an index number: 1
```

```
The index 1 of the list is:
12
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 2
Enter an index number: 6
The index 6 of the list is:
96
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 4
Enter the list number: 2
Enter an index number: 8
The index 8 of the list is:
 out of bound
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice:
```

Sample run 4

```
Enter the name of the course: CS305
Course CS305 registration list
Please enter a student ID !
12
Please enter a student ID!
12
Please enter a student ID!
45
Please enter a student ID!
45
Please enter a student ID!
85
Please enter a student ID!
```

```
65
Please enter a student ID!
32
Please enter a student ID!
^Z
List of students who registered for course CS305 is:
12
32
45
65
85

Enter the name of the course: CS405
Course CS405 registration list
Please enter a student ID !
^Z
List of students who registered for course CS405 is:
List is empty

****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 1
Union of two lists are:
12
32
45
65
85
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************

Your choice: 2
Enter the lhs list: 1

difference of list1-list2 is:
12
32
45
65
85
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
```

```
*  4. [] operator         *
*  5. Exit              *
****************************

Your choice: 2
Enter the lhs list: 2

difference of list2-list1 is:
List is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit              *
****************************

Your choice: 3
Intersection of list1 and list2 is:
List is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit              *
****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 3
The index 3 of the list is:
65
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit              *
****************************

Your choice: 4
Enter the list number: 2
Enter an index number: 4
The index 4 of the list is:
The list is empty
****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit              *
****************************
Your choice:
```

Sample run 5

```
Enter the name of the course: CS202
Course CS202 registration list
Please enter a student ID !
12
Please enter a student ID!
12al
Please enter a student ID!
45cg
Please enter a student ID!
8567
Please enter a student ID!
hj56
Please enter a student ID!
^Z
List of students who registered for course C
12
12al
45cg
8567
hj56

Enter the name of the course: CS301
Course CS301 registration list
Please enter a student ID !
25
Please enter a student ID!
45ds
Please enter a student ID!
12al
Please enter a student ID!
hj56
Please enter a student ID!
58kj
Please enter a student ID!
LJ47
Please enter a student ID!
^Z
List of students who registered for course C
12al
25
45ds
58kj
LJ47
hj56

****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
****************************
```

```
Your choice: 1
Union of two lists are:
12
12al
25
45cg
45ds
58kj
8567
LJ47
hj56
*****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
*****************************

Your choice: 2
Enter the lhs list: 1

difference of list1-list2 is:
12
45cg
8567
*****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
*****************************

Your choice: 2
Enter the lhs list: 2

difference of list2-list1 is:
25
45ds
58kj
LJ47
*****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
*****************************

Your choice: 3
Intersection of list1 and list2 is:
12al
hj56
```

```
*****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
*****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 3
The index 3 of the list is:
8567
*****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
*****************************

Your choice: 4
Enter the list number: 1
Enter an index number: 0
The index 0 of the list is:
12
*****************************
*  1. union(+)           *
*  2. difference(-)      *
*  3. Intersection (&)   *
*  4. [] operator        *
*  5. Exit               *
*****************************
```

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 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

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 (LeyliJavidKhayati)