

Course: BBM203 Fall  
Name: Assignment 2  
TA: Hayriye Çelikbilek  
Given: 16<sup>th</sup> November 2021  
Due: 30<sup>th</sup> November 2021

# Linked List

## Programming Language: C++

### 1. Introduction

Check for BBM201 and BBM203 course sources for linked lists, classes and headers.

### 2. Problem Statement

With the object-oriented programming method, an "Employee Recording System" application will be developed for a public institution. Quick access to the following employee data is expected via the employee number and date of appointment to the institution.

- Employee number: 4-digit integer
- Employee type: temporary or permanent (Obtained from user 0 as temp. 1 as perm.)
- Name: 12 characters
- Surname: 12 characters
- Title: 12 characters
- Salary coefficient: real number
- Date of birth: Date (separate integers for day, month, and year)
- Date of appointment to the institution: Date (separate integers for day, month, and year)
- Length of service in other institutions (days): integer (will not be asked if not necessary)

To access employee information via employee number, a circular single-linked list with a top-of-list node will be used, in ascending order by employee number. A double-linked list with a top-of-the-list node will be used to access employee information via the date of appointment to the institution. There are two types of employee subclasses: TemporaryEmployee and

PermanentEmployee. Because TemporaryEmployee is not many in numbers but frequently changing in the system, it will be an array implementation (Maximum number of temporary employees are 20). Similarly, since the majority of employees are PermanentEmployee, it will be implemented with dynamic memory allocation. The following table summarizes the structure.

	TemporaryEmployee	PermanentEmployee
<b>Linked List Type</b>	Circular single linked list	Double linked list
<b>Implementation Type</b>	Array Implementation	Dynamic Memory Allocation
<b>Inheritance</b>	Employee's subclass	Employee's subclass
<b>Order by</b>	Employee number	Date of appointment
<b>Unique access by</b>	Employee number	Employee number

Accordingly, the requests listed below are expected to be fulfilled via a menu offered in the console application:

- 1) Appointment of an employee, who started working for the first time, to the institution  
First, the employee number must be obtained from the user and if there is an employee with the same number, no addition should be made, otherwise the other information of the employee (starting from the employee type in the given order) must be obtained from the user and the insertion must be performed.
- 2) Appointment of an employee, who transferred from another institution, to the institution  
First, the employee number must be obtained from the user and if there is an employee with this number, no addition should be made, otherwise the other information of the employee (starting from the employee type in the given order) must be obtained from the user and the insertion must be performed.
- 3) Updating the title and salary coefficient of an employee  
First, the employee number must be obtained from the user, and if there is an employee with this number, the title and salary coefficient must be updated.
- 4) Deletion of an employee  
First, the employee number must be obtained from the user, and if there is an employee with this number, the deletion process must be performed.
- 5) Listing the information of an employee  
First, the employee number must be obtained from the user, and if there is an employee with this number, the information must be listed.
- 6) Listing of all employees in the order of employee number  
All employees' information should be listed in the ascending order by employee number.

- 7) Listing of all employees in the order of their appointment to the institution  
The information of all employees should be listed in order from oldest to newest according to the date of appointment to the institution.
- 8) Listing the employees assigned to the institution after a certain date  
First, the date should be taken from the user, and the information of the all employees assigned to the institution after this date should be listed in order from new to old according to the date of appointment to the institution.
- 9) Listing the employees assigned to the institution in a given year  
First, the year must be taken from the user and the information of the all employees assigned to the institution in this year should be listed in order from oldest to newest according to the date of appointment to the institution.
- 10) Listing of employees born before a certain date  
First, the date should be taken from the user and the information of all employees born before this date should be listed in ascending order according to the employee number.
- 11) Listing of employees born in a particular month  
First, the month should be taken from the user and the information of all employees born in this month should be listed in ascending order according to the employee number.
- 12) Listing the information of the employee with a certain title, who was last assigned to the institution  
First, the title should be taken from the user, and the information of the employee with this title, who was last assigned to the institution, should be listed.

### 3. Requirements

- 1) Assume that data entries will be made correctly, do not check for errors.
- 2) Make sure that your algorithm is efficient in search operations.
- 3) The project will be implemented, build and run in the dev environment.
- 4) The project will be delivered to the submit system of our department.
- 5) In terms of object-oriented programming principles, the project is expected to have the following characteristics:
  - a. At least Employee, TemporaryEmployee, PermanentEmployee, Date, CircularArrayLinkedList and DoubleDynamicLinkedList classes must be defined. The purpose of list classes is to hold the top-of-the-list node for the linked list and to offer operations (add, delete, search/find, list/query, etc.) on the list. Having a single list class will result in loss of points.
  - b. Data members of classes must be specified as private.

- c. Data members (except arrays) that are not expected to change must be specified as `const`.
- d. Classes should have the necessary read and set (`get&set`) functions.
- e. Constructor functions should use default arguments if appropriate.
- f. The `<<` operator must be overloaded for the `Employee`, `TemporaryEmployee`, `PermanentEmployee`, `Date`, `CircularArrayLinkedList` and `DoubleDynamicLinkedList` classes.
- g. Comparison operators (`<`, `>`, `<=`, `>=`, `==` etc.) required for the date class must be overloaded with the member function method.
- h. Interfaces (header files) and implementations (cpp files) of classes must be separate.
- i. Interfaces and their parts must be clearly explained in the report.

## 6. Inputs and Outputs

The console application will look like the following when it is started. It will keep turning back to this menu after operationg the selected operations. The system will be on unless the user closes the application (your executable program)

```
____ Employee Recording System ____
Please select for the following Menu Operation:
1) Appointment of a new employee
2) Appointment of a transferred employee
3) Updating the title and salary coefficient of an employee
4) Deletion of an employee
5) Listing the information of an employee
6) Listing employees ordered by employee number
7) Listing employees ordered by appointment date
8) Listing employees appointed after a certain date
9) Listing employees assigned in a given year
10) Listing employees born before a certain date
11) Listing employees born in a particular month
12) Listing the information of the last assigned employee with a given title
```

Example of the second operation given below. The following console image inserting a new employee to the system with the whole necessary information. Notice the date terms are separated with "-" character and the length of service is necessary for only the transferred employees (will not be asked to the user for new employees, default 0)

```

_____ Employee Recording System _____
Please select for the following Menu Operation:
1) Appointment of a new employee
2) Appointment of a transferred employee
3) Updating the title and salary coefficient of an employee
4) Deletion of an employee
5) Listing the information of an employee
6) Listing employees ordered by employee number
7) Listing employees ordered by appointment date
8) Listing employees appointed after a certain date
9) Listing employees assigned in a given year
10) Listing employees born before a certain date
11) Listing employees born in a particular month
12) Listing the information of the last assigned employee with a given title
2
Please type the employee number
[5555
Type the employee type
1
Type the Name
[Hayriye
Type surname
[Celikbilek
Type title
[TA
Type salary coefficient
[5.6
Type birth date
[01-01-1980
Type appointment date
[01-01-2020
Type length of service days
[50

```

## 7. Grading

Evaluation Metric	Points	Your guess/expectation of grading (evaluate your self in your report)
Efficiency of the program	5	...
Compliance with OOP principles	25	...
Correctly working parts	45	...
Report	25	...

### Report Format (Required)

- 1) Cover: Your name, student number, course name, delivery date.
- 2) Introduction: Define the project in your own words. What is the problem description?  
Project goals?

- 3) Method: What are your methods to solve the problem? Why?
- 4) Development
  - 1) Plan: What is the project plan? Your requirements.
  - 2) Analysis: What kind of analysis have you done? Why? State the problem in your own words.
  - 3) Design: What is your solution overflow? Why? Which methods have you chosen? Why? What are their functions/properties? Why are they beneficial to our problem? Which other methods may be chosen? Why are they eliminated? How do you tune your solution? Write down your thoughts on the solution of the problem and write down the interfaces of the abstract data types (classes) required for the solution. Explain your design.
  - 4) Implementation: Flow of the implementation. Explain your code parts such as functions, selected libraries/methods.
  - 5) Programmer Catalog: Write down the time you spend for analysis, design, implementation, testing, and reporting. Can other developers reuse your code/solution? When? How? What are your suggestions to other programmers? Add the source code parts here in the report and explain them. Include your header files and describe the data members and member functions of the abstract data types (classes) in your project. This part is the manual for programmers. Explain your design details.
  - 6) User Catalog: How can one use your code for other purposes / for other data? Prepare the user manual of the program (might use screenshots). State the restrictions, if any—your suggestions to users. This part is the user guide of the program.
- 5) Results, Discussion and Conclusion: Your program results. Which methods have you used? Why? Visual statements are preferable. Evaluation and comments might be listed. Discuss your results and conclude with several statements about several aspects of your solution. You may use whatever is necessary to explain your statements. Your grading to your project.
- 6) References: Resources you have used to prepare the project and the report. Use numbers that are indicated in the text.

## 8. Notes

- Do not miss the deadline.
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- Write READABLE SOURCE CODE blocks.

- You can ask your questions via Piazza (<https://piazza.com/hacettepe.edu.tr/fall/bbm203>) and you are supposed to be aware of everything discussed in Piazza.
- You will use online submission system to submit your experiments. <https://submit.cs.hacettepe.edu.tr/>
- No other submission method (email or etc.) will be accepted. Do not submit any file via e-mail related with this assignment.
- File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system). You must submit your work with the file hierarchy stated below:
  - ➡ <student\_id.zip>/(Required)
    - ➡ src/(Required)
      - ➡ Main.cpp(Required)
      - ➡ \*.cpp(optional)
      - ➡ \*.h(optional)
    - ➡ Report.pdf(Required)

## 9. Policy

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work (from internet), in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.