CS-303 Assignment 2

(25 points)

1. (15 points) Build a Single_Linked_List class. Your class should have the data members: head, tail, and num_items. Write the following member functions, which perform the same operations as the corresponding functions in the standard list class:

- push_front,
- push_back,
- pop_front,
- pop_back,
- front, back,
- empty,
- void insert(size_t index, const Item_Type& item): Insert item at position index (starting at 0). Insert at the end if index is beyond the end of the list
- bool remove(size_t index): Remove the item at position index. Return true if successful; return false if index is beyond the end of the list.
- size_t find(const Item_Type& item): Return the position of the first occurrence of item if it is found. Return the size of the list if it is not found.

2. (10 points) A company has two different kinds of employees: professional and nonprofessional. Generally, professional employees have a monthly salary, whereas nonprofessional employees are paid an hourly rate. Similarly, professional employees have a certain number of days of vacation, whereas nonprofessional employees receive vacation hours based on the number of hours they have worked. The amount contributed for health insurance is also different for each kind of employee.
    a. Use an abstract class Employee to store information common to all employees and to declare member functions for calculating weekly salary (number of hours worked * hourly rate) and computing health care contributions and vacation days earned that week (come up with some rules for these 2) . You need 3 functions (calculating weekly salary, health care contributions, and Vacation days).
    b. Define derived classes Professional and Nonprofessional. Create instance for them.
    c. Test your classes.

Submission guidelines:

1. You should have a header file and .cpp file. The header file should provide the function declaration and .cpp file should have implementation details.
2. All the functionality of the program should be implemented as functions and methods.
3. The code should be well commented
4. Create a report (readme file) that contains instruction on how to run the code and screen shots of the outputs
5. Upload your report and code files to GitHub.
6. Submit the GitHub link on Canvas by due date.