University of Newcastle
Discipline of Computer Science and Software Engineering
Semester 2, 2016 - SENG1120/6120

Assignment 1 – Version 3 (23 Aug)

Due using the Blackboard Assignment submission facility:
11:59PM – Friday, 09 September 2016

NOTE: *The important information about submission and code specifics at the end of this assignment specification***.**

INTRODUCTION

A situation exists by which your client (me) requires the ability to represent sets of playing cards. The name of the class that will do that should be `DeckOfCards`, and its internal data structure will be a linked list, whose nodes will store `string` objects.

ASSIGNMENT TASK

You are required to use a linked list, as discussed in lectures, to create your own implementation of the `DeckOfCards` class. You will initially implement a class called `LinkedList` that uses instances of `Node` to store instances of `value_type` (in this assignment, each `Node` will be used to store a single playing card represented by a `string` object). A `DeckOfCards` object will contain a private instance of `LinkedList` and provide the necessary wrapper code to pass information into and out of the list.

Your `DeckOfCards` class will implement the following member functions:

- Constructor, which creates a full deck of cards (numbers 2 to 10, plus J, Q, K, A), i.e. 13 ranks; four cards of each suit, i.e. “S” for spades, “H” for hearts, “C” for clubs, “D” for diamonds). For example “Q-S” is a Queen of spades, “10-H” is a 10 of hearts.

  - When printed, the initial sequence should show: 2-S 3-S 4-S … J-S Q-S K-S A-S 2-H 3-H … K-H A-H 2-C 3-C … K-C A-C 2-D 3-D … K-D A-D

- `void shuffle()` randomizes the sequence of cards. This method should use the Yates algorithm (https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle).
- `int length()` returns the number of cards in `DeckOfCards`.
- `bool empty()` returns whether there are any cards in `DeckOfCards` or not.
- `int position(string)` returns the position in the list with the first occurrence of the input parameter, -1 if the card is not in the deck.
- `string value()` returns a string displaying the sequence of cards stored in `DeckOfCards`.
- Overloaded output operator (i.e. ‘<<’) outputs the content of the `DeckOfCards` in a form suitable for printing. Hint: This method should use `string value()` in its implementation.
- `bool remove(string)` takes a single `string` argument and removes the corresponding card in `DeckOfCards`.

<u>SENG6120 students should implement, in addition to the previous member functions, the following one:</u>

- `reverse()` reverses the order of the nodes in `DeckOfCards`, which will cause the output to reverse as well.

DEMO PROGRAM

To demonstrate your `DeckOfCards` class you will write a program called `DeckOfCardsDemo` that uses your `DeckOfCards` class in the following way. When executed, `DeckOfCardsDemo` will:

1. create a new instance of `DeckOfCards` storing a full deck of cards
2. Print the deck of cards using `value()`. The card values have to be separated by a single blank space, all in the same line.
3. Shuffle the deck of cards using `shuffle()`
4. Print the deck of cards again using `cout <<`
5. Print the position of cards “4-H”, “10-S”, “Q-C” and “A-D”, separated by a single blank space, all in the same line.
6. Print the total number of cards in `DeckOfCards` using `length()`
7. Remove the cards “4-H” and “10-S”
8. Print the position of cards “4-H”, “10-S”, “Q-C” and “A-D” again
9. Print the total number of cards in `DeckOfCards` using `length()`

SENG6120 students should implement, in addition to the previous demo steps, the following one:

10. DeckOfCardsDemo will reverse DeckOfCards and print its content
    - You need to use the function reverse() and the overloaded << operator.

We will provide a version of DeckOfCardsDemo.cpp with a special main method declaration and that seeds the random number generator. Make sure you use it to implement steps 1-9(10).

The method to be used for shuffling the deck of cards has to be the Yates algorithm: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle

The pseudo code is as follows:

```
-- To shuffle an array a of n elements (indices 0..n-1):
for i from n−1 downto 1 do
     j ← random integer such that 0 ≤ j ≤ i
     exchange a[j] and a[i]
```

*** More information will be provided during the lectures. ***

Your submission should be made using the Assignments section of the course Blackboard site. **Incorrectly submitted assignments will not be marked. Assignments that do not use the specified class names will not be further marked.** You should provide all your .h and .cpp files, and a Makefile. Also, if necessary, provide a readme.txt file containing any instructions for the marker. Each program file should have a proper header section including your name, course and student number, and your code should be properly documented.

Remember that your code should compile and run correctly using Cygwin. There should be no segmentation faults or memory leaks during or after the execution of the program.

Compress all your files, including the cover sheet, into a single .zip file and submit it in by clicking in a link that I will create in the Assignments section on Blackboard especially for that.

Late submissions are subject to the rules specified in the Course Outline. Finally, a completed Assignment Cover Sheet should accompany your submission.

This assignment is worth 15 marks of your final result for the course.