CSCE 221 Cover Page
Programming Assignment #1

First Name:Abdullah Last Name: Ahmad UIN:927009064

User Name:abdullah2808 E-mail address:abdullah2808@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more in the Aggie Honor System Office http://aggiehonor.tamu.edu/

Type of sources	
People	
Web pages (provide URL)	https://www.tutorialspoint.com/cplusplus/input_output_operators_overloading.htm https://stackoverflow.com/questions/23504801/doubly-linked-list-implementation-c https://www.geeksforgeeks.org/data-structures/linked-list/ https://codereview.stackexchange.com/questions/125955/my-first-implementation-of-a-linked-
Printed material	
Other Sources	

I certify that I have listed all the sources that I used to develop the solutions/code to the submitted work.

"*On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*"

Your Name (signature) Abdullah Ahmad Date 6/6/2020

The Programming Assignment Report Instructions
CSCE 221

1. The description of an assignment problem.

   Throughout this assignment I constructed a record class, used starter code to construct a simple double linked list, a double linked list, and a templated double linked list. In the final part of the assignment I combined the templated double linked list with the book record class from part 0 to construct a library managment system.

2. The description of data structures and algorithms used to solve the problem.

   (a) Provide definitions of data structures by using Abstract Data Types (ADTs)

   Throughout the programming assignment linked lists were implemented. A linked list keeps track of both the previous and the next node in the list.

   (b) Write about the ADTs implementation in C++.

   To implement a linked list in C++ two classes were used the DLList class and the DLListNode class, one to construct nodes and one to organize the nodes. Furthermore, the classes were then worked on further and turned into templated classes such that any type of data type could be inserted such as ints, strings, or doubles.

   (c) Describe algorithms used to solve the problem.

   In the record class first the members were assigned (Author, Title, Edition, ISBN, and Year). After the members were assigned then the operators were overlaoded (>> , <<, <). The >> is used to read in data line by line, the << outputs the record, and the < compares the record to other records. The time complexity of all the operators is O(n) = 1, because they don't have any loops and all take a constant time to run that isn't dependent on the amount of iterations. Furthermore, in part 1 multiple different types of double linked lists were created, I will discuss the time complexity for the templated double linked list and 7 functions implemented throughout it since this has the overall same code structure as the other double linked list implemented but it furthers the usability of the code. The insert_before, insert_after, remove_before, remove_after, remove_first, and remove_last all have a constant time complexity of O(n) = 1, but their running time varies based on the amount of instructions within the function. For the insert_before and insert_after the running time is f(n) = 6, (I'm considering all assignment statements as 1 and the if statement as a 1). For the remove_before, remove_last, remove_first, and remove_after the running time is f(n) = 7. Furthermore for the function << the running time is f(n) = 2n + 1 (2 is the amount of functions inside the while loop and 1 is the assignment before the loop), the Big-O is O(n)=n. In part 2 a search function was implemented along with a read and insert function one to read data from a list and another to insert it by organization of Title into a linked list. The time complexity for the search function is f(n) = 6 + 4n + 4n + 6 (the two 6s are assign statements used before the two while loops and the 4 is in the amount of instructions within the loop), the Big-O is O(n) = n. The search function has two loops one to search the vector index of the corresponding first letter of the search input by title, and if more than 1 title is found then the second loop runs and searchs for a matching edition. The insert function has a constant time so Big-O is 1 and the running time is 9. The read function has a linear time of n, since the while loop runs until the input file has been read entirely, the running time is f(n)= 4+5n.

   (d) Analyze the algorithms according to assignment requirements.

   The analysis is done above in the description of the algorithms.

3. A C++ organization and implementation of the problem solution

   (a) Provide a list and description of classes or interfaces used by a program such as classes used to implement the data structures or exceptions.

   Classes used Record, DLList, DLListNode. The Record class has the functions << >> and < and the members Title, Author, Edition, ISBN, and Year as strings. The DLList class has different functions to operate a linked list, and the DLListNode constructs the nodes used throughout the DLList class.

(b) Include in the report the class declarations from a header file (.h) and their implementation from a source file (.cpp).

```
class Record {
public:

    friend istream& operator>>(istream& input, Record& R) { //input operator takes in books fr
        string line;
        getline(input, line);
        if (line.size() == 0)  getline(input, line);
        R.Title = line;
        getline(input, line);
        R.Author = line;
        getline(input, line);
        R.ISBN = line;
        getline(input, line);
        R.Year = line;
        getline(input, line);
        R.Edition = line;
        return input;
    } // Since there is 13 operations in the function the running time is 13, and the Big-Nota
      // So the total runtime in the scope of the program is 130.
    friend ostream& operator<<(ostream& output, const Record& R) { //output operator
        output << "Title: " << R.Title << " Author: " << R.Author << " ISBN: " << R.ISBN << "
            << " Edition: " << R.Edition;
        return output;
    } // Since there is 10 cout operations in the function the running time is 10, and the Big
      // So the total runtime in the scope of the program is 100.

    friend bool operator<(const Record& r1, const Record& r2) {
        return (r1.Title == r2.Title) && (r1.Author == r2.Author) && (r1.ISBN == r2.ISBN);
    }



    string ISBN, Title, Author, Edition, Year;
};
```

```cpp
template<class T>
struct DLListNode {
    T obj;
    DLListNode<T> *prev, * next;
    // constructor
    DLListNode<T>(T e = T(), DLListNode<T>* p = nullptr, DLListNode<T>* n = nullptr)
        : obj(e), prev(p), next(n) {}
    T getElem() const { return obj; }
    DLListNode<T>* getNext() const { return next; }
    DLListNode<T>* getPrev() const { return prev; }
};

// doubly linked list class
template <class T> class DLList {  <T> Provide sample template arguments for IntelliSense
private:
    DLListNode<T> header, trailer;
public:
    DLList() : header(T()), trailer(T()) // default constructor
    {
        header.next = &trailer; trailer.prev = &header;
    }
    DLList<T>(const DLList<T>& dll); // copy constructor
    DLList<T>(DLList<T>&& dll); // move constructor
    ~DLList<T>(); // destructor
    DLList<T>& operator=(const DLList<T>& dll); // copy assignment operator
    DLList<T>& operator=(DLList<T>&& dll); // move assignment operator
    // return the pointer to the first node
    DLListNode<T>* first_node() const { return header.next; }
    // return the pointer to the trailer
    const DLListNode<T>* after_last_node() const { return &trailer; }
    // return if the list is empty
    bool is_empty() const { return header.next == &trailer; }
    T first() const; // return the first object
    T last() const; // return the last object
    void insert_first(T obj); // insert to the first node
    T remove_first(); // remove the first node
    void insert_last(T obj); // insert to the last node
    T remove_last(); // remove the last node
    void insert_after(DLListNode<T>& p, T obj);
    void insert_before(DLListNode<T>& p, T obj);
    T remove_after(DLListNode<T>& p);
    T remove_before(DLListNode<T>& p);
    DLListNode<T>* insertOrdered(const T& obj);
};
```

(c) Provide features of the C++ programming paradigms like Inheritance or Polymorphism in case of object oriented programming, or Templates in the case of generic programming used in your implementation.
Templates were used to offer generic programming so that any data type can be utilized.

4. A user guide description how to navigate your program with the instructions how to:

(a) compile the program: specify the directory and file names, etc.
I used visual studio to compile and run the program, I added everything to the project and when into file properties to run the files I wanted and disable the other files from being compiled and ran.

(b) run the program: specify the name of an executable file.
I used visual studio to run the programs.

5. Specifications and description of input and output formats and files

   (a) The type of files: keyboard, text files, etc (if applicable).
   The book.txt file is used and the keyboard is used to insert search inputs.

   (b) A file input format: when a program requires a sequence of input items, specify the number of items per line or a line termination. Provide a sample of a required input format.
   The given book.txt file that was given to use for the class record is the way the data should be setup in a text file.

   (c) Discuss possible cases when your program could crash because of incorrect input (a wrong file name, strings instead of a number, or such cases when the program expects 10 items to read and it finds only 9.)

   The search function cannot handle lower cases letters inserted as the first letter of the search neither can the text input file, because of the index going less than 0 when inserting into the vector of linked lists.

6. Provide types of exceptions and their purpose in your program.

   (a) logical exceptions (such as deletion of an item from an empty container, etc.).
   Throughout my program I used the exception starter code given at the top to be thrown whenever the linked list is empty.

   (b) runtime exception (such as division by 0, etc.)

7. Test your program for correctness using valid, invalid, and random inputs (e.g., insertion of an item at the beginning, at the end, or at a random place into a sorted vector). Include evidence of your testing, such as an output file or screen shots with an input and the corresponding output.

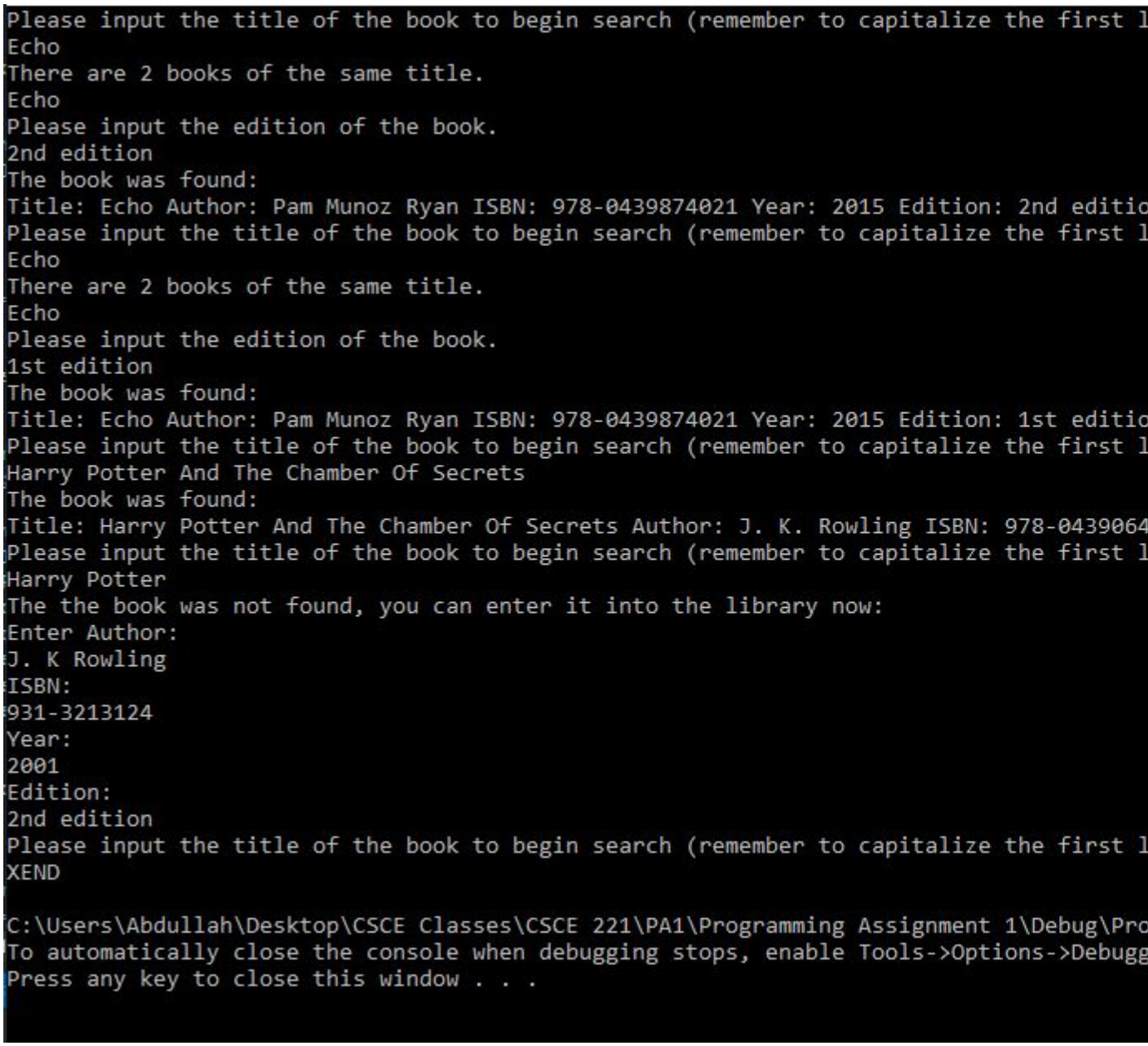
   Class record:

```
Title: Harry Potter And The Chamber Of Secrets Author: J. K. Rowling ISBN: 978-0439064
Title: H is for Hawk Author: Helen Macdonald ISBN: 978-0802123411 Year: 2015 Edition:
Title: Harry Potter and the Cursed Child Author: J. K. Rowling ISBN: 978-1338099133 Ye
Title: The Kite Runner Author: Khaled Hosseini ISBN: 978-1594631931 Year: 2013 Edition
Title: Red Sorghum: A Novel of China Author: Mo Yan ISBN: 978-0140168549 Year: 1994 Ed
Title: Pride and Prejudice Author: Jane Austen ISBN: 978-7020040179 Year: 1993 Edition
Title: Harry Potter and the Prisoner of Azkaban Author: J. K. Rowling ISBN: 978-043913
Title: The War that Saved My Life Author: Kimberly Brubaker Bradley ISBN: 978-08037408
2015 Edition: 1st edition
Title: Echo Author: Pam Munoz Ryan ISBN: 978-0439874021 Year: 2015 Edition: 2nd editio
Title: Artificial Intelligence: A Modern Approach Author: Stuart Russell, Peter Norvig
```

   Tempated Double Linked List:

```
Insert 10 nodes at front with value 10,20,30,..,100
list: 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100

Copy to a new list
list2: 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100

Assign to another new list
list3: 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100

Delete the last 10 nodes
list: 100 90 80 70 60 50 40 30 20 10

Delete the first 10 nodes
list:

Make sure the other two lists are not affected.
list2: 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
list3: 100 90 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
Testing insert after on list 2
inserting after the value 90
and inserting a string hello
100 90 hello 80 70 60 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100

Testing insert before on list 3
inserting before the value 50
and inserting a string blue
100 90 80 70 60 blue 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100

Testing remove after on list 2
Removing after 40
100 90 hello 80 70 60 50 40 20 10 10 20 30 40 50 60 70 80 90 100

Testing remove before on list 3
Removing before blue
100 90 80 70 blue 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100

testing move assignment by creating new list 4 by moving list 2 into it
list 4:
100 90 hello 80 70 60 50 40 20 10 10 20 30 40 50 60 70 80 90 100

testing copy assignment by creating new list 5 by copying list 3
list 5:
100 90 80 70 blue 50 40 30 20 10 10 20 30 40 50 60 70 80 90 100
```

Library Managment:


Please input the title of the book to begin search (remember to capitalize the first l
Echo
There are 2 books of the same title.
Echo
Please input the edition of the book.
2nd edition
The book was found:
Title: Echo Author: Pam Munoz Ryan ISBN: 978-0439874021 Year: 2015 Edition: 2nd editio
Please input the title of the book to begin search (remember to capitalize the first l
Echo
There are 2 books of the same title.
Echo
Please input the edition of the book.
1st edition
The book was found:
Title: Echo Author: Pam Munoz Ryan ISBN: 978-0439874021 Year: 2015 Edition: 1st editio
Please input the title of the book to begin search (remember to capitalize the first l
Harry Potter And The Chamber Of Secrets
The book was found:
Title: Harry Potter And The Chamber Of Secrets Author: J. K. Rowling ISBN: 978-0439064
Please input the title of the book to begin search (remember to capitalize the first l
Harry Potter
The the book was not found, you can enter it into the library now:
Enter Author:
J. K Rowling
ISBN:
931-3213124
Year:
2001
Edition:
2nd edition
Please input the title of the book to begin search (remember to capitalize the first l
XEND
C:\Users\Abdullah\Desktop\CSCE Classes\CSCE 221\PA1\Programming Assignment 1\Debug\Pro
To automatically close the console when debugging stops, enable Tools->Options->Debugg
Press any key to close this window . . .