CS201-Data Structures
Fall 2018
Assignment 2
Maximum Marks: 10
Due: 21-OCT-2018
Assignment Submission Soft on slate

Coding assignment: You have to code all the question given below in C++.

Grade Point: This is a graded assignment of 10 points

Individual Assignment: This is an individual assignment

Submission time: **Sunday 21-OCT-2018 11:55 PM**

Submission type: Soft copy on slate.

Submission requirements:

- Submit all the code on slate before deadline.
- A presentation or viva-voce could be taken if required.
- **No assignment will be accepting on email.**

Submit as

1- Well arrange C/C++ source files. (error-free ready-to-run soft copies).
2- The program MUST have a good menu-driven user-friendly interface.
3- Soft copy of the text files **Books1.txt**. only for question 1 **(use simple file handling)**
4- Make a single zip/compressed file of all the individual files and upload it on Slate.

Question 1

Consider a library database comprising of books records with the following attributes:

- ***Book_ID*** (auto generated ID, e.g. 1240 **unique**)
- ***Book _Title*** (maximum of 30 characters)
- ***Author_Name*** (maximum of 30 characters; consider single/first author only)
- ***Status*** (*issued/not issued/Lost*. **Hint**: you can use a flag variable, with three possible values, for this)
- ***Date_of_Issue_or_Returned*** (if *issued*, shows that date of issue; if *not issued*, it shows the date of last returned; finally, if *missing*, it shows the date since it went missing. **Format**: ddmmyyyy.)

Your job is to write a C/C++ code to be able to perform the following operations on the book database. **You are required to use a single-linked list for this program.**

1. Enter data for at least five records of this entity into memory. You are required to program a **static array** in C/C++ to read this particular data into memory. Store data for five books, using assignment statements in order to save time.
2. Next, open a text file **Books1.txt** using file streams in C/C++ and write the above data from memory into a permanent storage device.
3. Now, given the data is permanently stored into a file, implement C/C++ routines which should be able to perform the following functions:

a. Read the text file **Books1.txt** using C/C++ file streams, and temporarily load this data into a linked-list ***L1*** with the header named as ***BooksHead*1**.

b. Once the data is read into the linked list, the program should prompt the user to perform the following tasks through a menu:

i. Display all the books' data in the list in tabular form. The tabular form means all the data related to one book should be in one row (or two rows at most, if the size of the record exceeds the rows size).

ii. Append a new book at the end of the list.

iii. Search and delete a particular book's record form the list. (The book information will be searched by giving the full ***Book_Title or Book_ID***. In case, no book with that name is found, an error message will be displayed on the screen.)

iv. The title of the book should be unique, if the book already exists then the system return with a msg by displaying all the information of book which you want to insert and it is present in the database.

In order to give you a jump start, two separate cpp files have to make and upload.

1- The first file named **linklist.cpp** implements a simple linked list system. You can easily modify/extend it for the library database. Moreover, this program should give the implementation of **append** and **display** functions for the simplified linked list. Also make inset delete and search module.
2- The second file **FileHandling.cpp** provides a simplified example of reading/ writing a list of data from/to a text file using C FILE STREAM (You are free to use C++ IOSTREAM if you are more comfortable with that). Moreover, notice that this example uses **struct** arrays to temporarily hold the data in memory, both before writing to the text file and after reading from the text file. In the term project, the static array is only used to store the first set of books. The linked-list is used for temporary storage of data in memory for the rest of the program.

   However, it is advisable that if you are struggling with any part of the project, then first complete other parts (which seem easier to you) and complete difficult part later on weekend.

Question 2

Implement the C++ program using stack concepts to evaluate parenthesis of an expression given by the user in infix form, if the parenthesis is placed right then convert this infix expression to postfix and also to prefix. Make sure your program show the values present in stack at each step. Example of valid and invalid expression are given below

Valid `[{((B + C) / 3 - 47 % E) * (F + 8)}]`
Invalid `{[((B + C) / 3 - 47 % E) * (F + 8)}]`

Question 3

Design C++ program or use the program of Q.2 to evaluate the expression given in infix form and display the result using the concepts of stacks.

Question 4

Provide link list based implementation of stack.

Question 5

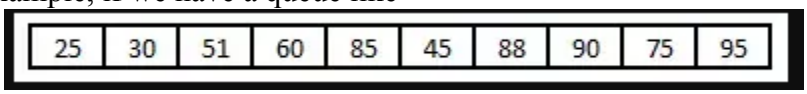
Complete the following code to implement the queue in class with all function definitions

```
class DSqueue
{
      int array1[10];
      int front_end, rear_end;
public :
      DSqueue (){front_end = - 1; rear_end= - 1 }
      void Enqueue();
      void Dequeue();
```

```
        bool isempty();
        bool isfull()
}
```

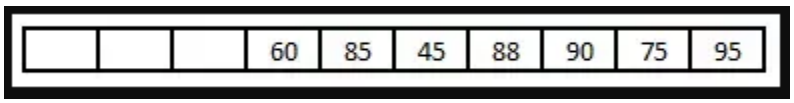
Question 6

In a queue data structure, elements are enqueue into the queue until the queue's capacity is reached. However, insertion take place when there is some location free after the read pointer.
Let take an example, if we have a queue like



In given scenario we can't insert values to this queue.

Now, if the first three members are de-queued from the front (left hand side) of the queue, we get:



Where the queue remains full but we can not insert a new element because, the back of the queue (right hand side) remains as it was before. this is the major limitation of a classical queue, i.e. even if there is space available at the front of the queue we can not use that space to add values.
You have to design a mechanism and **implement it in C++** to use the space free in queue (defined in array as given above not LinkList) for new item to insert.

Question 7

Provide link list based implantation of queue.