Mini-Projects DSSD -AUG - 2014

Project – I : (C or C++)

A Tool that analyzes a text file and returns information about its contents

1) we will use an in-order linked list to build an indexing service for file contents.

2) The program will be passed a file name and a command.

3) Each word in the file is inserted into the in-order list. If a word is encountered more than once, a counter for its frequency of occurrence is incremented.
4) The following actions will then be performed by the program:
i. command: wordCount -Prints the number of words in the file.
ii. command: distWords - Prints the number of distinct words in the file.
iii. command: charCount -Prints the number of characters in the file (including spaces)
iv. command: frequentWord - Prints the word(s) that had occurred the most in the input file, if two or more words have the same number of repetitions, it should print all of them.
v. command: countWord myword -Prints the number of myword occurrence in the file
vi. command: starting mysub -Prints all words that start with mysub followed by their number of occurrences.
Ex: if the program is passed a command starting He it shall print all words starting with He, for example Hello Hey.
vii. command : contains mystr -Prints all words that containing the string mystr followed by their number of occurrences.
Ex: if the program is passed a command contains al, it shall print words such as shall all.
viii. command: print Prints all words followed by their number of occurrences in their order of occurrences.

ERROR Handling :

You should check for the correctness of every command
(ex: number of arguments).
You may use the following error messages to reflect the reason of not
>performing the command
>File not found
>Undefined command
>Incorrect number of arguments

Project – II : (C or C++)

You are required to implement the following application using a C++ programming language:

There is a Bike Agency that contains many kind of Bikes and each client give an order to buy a bike. The bike agency employees will search for the required bike upon customers' request in order (First come First Served).

The bike agency system has the following requirements:

1) There are many kinds of bikes in the bike agency's inventory at least twenty. They may have more than one bike from each kind.

2) The customers are organized as a Queue.

3) Retrieve data of the last sold bike.

Your program should consist of the following:

1) A class implementing a linked list including the following functions

a) InsertBeginn b) InsertEnd

c) DeletefromBegin d) DeletefromEnd

e) DeleteSpecific

2) A class implementing a stack inherited from the linked list class and has the following functions

a) Push b) Pop c) Top

3) A class implementing a queue inherited from the linked list class and has the following functions

a) Enqueue b) Dequeue c) Front

4) Your classes should contain at least one constructor and a destructor.

The program flow is as follows:

The bike names are saved in a linked list. The customer orders are saved in a queue. You should take the bike order from the beginning of the queue and search for it in the linked list then delete it and put this sold bike in a stack to be able to retrieve the last sold bike.

Ex:

Menu :

1. Add a Bike
2. Book a Bike (Request)
3. Map customer to Bike
4. Last sold Bikes Details (In order)

Project – III : (C or C++)

Bank Management System

A Banker will be using the software.

The actions that can be performed by Banker are:

He should be

1) Able to create New Account for a customer.

2) Able to deposit amount into customers account

3) Able to withdraw amount from customers account

4) Able to enquiry Balance in customers account

5) Able to close account of a customer

6) Able to Modify account details of a customer

Note:

i. All the account details of the customer should be stored in file named acc_details.db file (If possible encrypt data by substitution method). (While displaying details Decrypt and display).

ii. All the transactions of all the customers should be stored in a Transaction_Register.db file (If possible encrypt data and store**) .**

Project IV : Dynamic Memory Manager in C++

Project V : Mini Search Engine

The search engine should be able to "quickly" find out all occurrences of a phrase in a set of 50 given text files written in English. Use any data structure and algorithm of your own choice.