**Lab 3 – List Class**

**CSC 3302**

|  |  |
| --- | --- |
|  |  |
| **Introduction:** | Creation and operation of linked lists using dynamic data structures provides flexibility in solving problems whose total number of components are not known until run-time. |
|  |  |
| **ADT List:** | The data organization for linked lists commonly consists of a class of problem specific information and a reference variable (this is commonly referred to as a self referential structure). Common operations for linked lists include **initializing** a linked list (constructors), **adding** elements, **deleting** elements, and **traversing** (printing) the list. The list should contain a pointer to the front of the list of elements. |
|  |  |
| **Problem:** | Write a program to maintain a stock inventory. This program must be able to add new components to the inventory, delete components, remove items from the stock, add items to the stock, and print an inventory list. |
|  |  |
| **Program:** | Write a program that reads in a series of commands (from a file) and manipulates an inventory list. The items should be placed in the inventory in the list ordered by component number. The program should read in a series of commands which will be either R, A, D, I, P, or Q. If the command is an “R”, then the number of units of the item for the component number specified should be reduced (and if zero, removed from the list with the appropriate message). If there is not an adequate number of components, an error message should be displayed. If the reorder level is reached, a reorder message should be displayed. If the command is an “A”, then the number of units of the item for the component number specified should be increased appropriately. If the command is a “D”, then the entire component should be removed from the inventory list and the node should be released for garbage collection. If the command is an “I”, a new stock item should be added using the specified component number (6 digits), description (up to 20 characters), initial number of units in stock (integer), and reorder level (integer) – one stock item per line. If the command is a “P”, then the entire list should be displayed. If the command is a “Q”, the program should display all inventory and quit.  OPERATION INPUT NEEDED  Remove from stock Component number, number of units removed.  Add to stock Component number, number of units added.  Delete component Component number.  Insert component Component number, description, units in stock, reorder level.  Print inventory None.  Exit program None.  The linked list should follow the class listed in the textbook (and discussed in class) modified to contain the above node (class) data type.  Name your main class p3.java, and name your List class InventoryList.java. You may put your node class in the same file as your List class. You must make your inventory list and nodes generic |
|  |  |
| **Input:** | A sample data set might be:  I 110010 STACK STRETCHER 10 2  I 101001 CIRCULAR QUEUE 15 1  I 666666 PSEUDOCODE GEN. 38 3  I 123456 LIST LINKER 11 5  I 567890 DOUBLE LISTER 12 4  D 101001  P  R 123456 4  R 123456 2  A 110010 10  I 1000001 TREE BARKER 8 3  P  D 666666  Q  Use **p3.dat** in the instructor’s account for the data. |
|  |  |

|  |  |
| --- | --- |
| **Output:** | Output will consist of appropriate messages and listing of inventory. |
|  | After performing any operation, the computer should confirm that the operation has been successfully performed. The operation "Remove from stock" should give rise to a warning message if the number of units in stock falls below the reorder level. Naturally, you cannot remove more from stock than the current number in stock. The operation "Print Inventory" should list the contents of the inventory in order of component number in a format similar to the following:  COMPONENT CURRENT # REORDER  NUMBER DESCRIPTION IN STOCK LEVEL  ----------------------------------------------------------------------------------------------  101001 CIRCULAR QUEUE 15 1  110010 STACK STRETCHER 0 2 \*REORDER\*  123456 LIST LINKER 11 5  666666 PSEUDOCODE GEN. 38 3 |
|  |  |
| **Hints:** | Build the linked list by reading and creating a list node (dynamic memory allocation), inserting the data items into the node, and then inserting the node in the appropriate place in the list. The list must consist of one entry for each different component in stock, each entry containing the component number, the description (up to 20 characters), the number of units in stock, and the reorder level for that component. The inventory is initially empty. In order to print the inventory in order, you will want to keep the list ordered at all times.  Write a separate function (method) for each ADT list operation (i.e., constructor(s), insert, delete, traverse (print), and search). Write a separate class for the node. |
| **Data:** | On the server, you will need to use the following absolute path and append the filename to it. "/home/courses/csci3302-002/datafiles/"  Remember that you must pass the data file name in as a command line argument. |