```
 1 /*************************************************************
 2 * AUTHOR     : Amirarsalan Valipour
 3 * STUDENT ID     : 1103126
 4 * Assignment #5  : DVD Movie ListIntro to OOP
 5 * CLASS          : CS 1B
 6 * SECTION        : MW - 7:30 pm - 9:50 pm
 7 * DUE DATE       : 12/16/2019
 8 *************************************************************/
 9
10 #ifndef STACKLIST_H_
11 #define STACKLIST_H_
12
13 #include "MyHeader.h"
14
15 class StackList     // Base Class -Using linked list implementation
16 {
17
18     public:
19
20         /**********************************
21          *   CONSTRUCTOR / DESTRUCTOR   *
22          **********************************/
23
24         StackList();
25
26         ~StackList();
27
28
29         /*******************
30          *    MUTATORS    *
31          *******************/
32
33         void Push(DVDNode newDVD); //create a DVDNode, add a
34                                    // DVDNode in the stack, by
35                                    // adding to the front of the
36                                    // linked List
37
38         DVDNode Pop();             // return the DVDNode in the top
   of
39                                    // the stack, remove the DVDNode
40                                    // from the stack, delete the
   DVDNode
41
42         /*******************
43          *    ACCESSORS    *
```

```
44          ********************/
45
46          bool IsEmpty() const;        // check if stack is empty
47
48
49          DVDNode Peek() const;        // return the DVDNode at the
  top
50                                       // of the stack
51
52          int Size() const;            // return the number of people
53                                       // in the stack
54
55      protected:
56
57          DVDNode *head;       // head pointer for stack
58
59          int stackCount;      // total number of persons in the stack
60  };
61
62  #endif /* STACKLIST_H_ */
63
64
65  /*******************************
66   *  CONSTRUCTOR & DESTRUCTOR  *
67   *******************************/
68
69  /*****************************************************************
70   * StackList ();
71   * Constructor; Initialize class attributes
72   * Parameters: none
73   * Return: none
74   *****************************************************************/
75
76  /*****************************************************************
77   * ~StackList ();
78   * Destructor; does not perform any specific function
79   * Parameters: none
80   * Return:     none
81   *****************************************************************/
82
83
84  /*************
85   * MUTATORS *
86   *************/
87
```

```
 88 /
    **********************************************************************
 89  * void Push (DVDNode newDVD);
 90  *
 91  * Mutator; This method will add a DVD node to the list to the
    front
 92  *
    ----------------------------------------------------------------
 93  * Parameter: newDVD (DVDNode) //IN - node to be added to list
 94  *
    ----------------------------------------------------------------
 95  * Return: none
 96
    **********************************************************************
    /
 97
 98 /
    **********************************************************************
 99  * DVDNode Pop ();
100  *
101  * Mutator; This method will remove a DVD node from the front of
    the
102  * list and return the DVDNode being removed
103  *
    ----------------------------------------------------------------
104  *  Parameter: none
105  *
    ----------------------------------------------------------------
106  * Return: dvdPtr (DVDNode)
107
    **********************************************************************
    */
108
109
110 /*************
111  * ACCESSORS *
112  *************/
113
114 /**********************************************************************
115  * bool IsEmpty () const;
116  *
117  * Accessor; This method will return the boolean value whether
118  * the list is empty or not empty
119  * ----------------------------------------------------------------
120  * Parameters: none
```

```
 * ---------------------------------------------------------------
 * Return: emptyCheck (bool)
 *****************************************************************/

/*****************************************************************
 * DVDNode Peek () const;
 *
 * Accessor; This method will return the DVD node of the first
 * element on the list
 * ---------------------------------------------------------------
 * Parameters: none
 * ---------------------------------------------------------------
 * Return: dvdPtr (DVDNode)
 *****************************************************************/

/*****************************************************************
* int Size () const;
*
* Accessor; This method will return the size of the list
* ---------------------------------------------------------------
* Parameters: none
* ---------------------------------------------------------------
* Return: stackCount (int)
*****************************************************************/

```