```
 1 /*****************************************************************************
 2  * PROGRAMMER : Ali Eshghi
 3  * STUDENT ID : 1112261
 4  * CLASS      : CS1B
 5  * SECTION    : MW 7:30pm
 6  * Assign #3  : Searching linked list
 7  * DUE DATE   : 6 November 2019
 8  *****************************************************************************/
 9 #ifndef CLASSHEADER_H_
10 #define CLASSHEADER_H_
11
12
13 #include<iostream>   // INPUT AND OUTPUT
14 #include<iomanip>    // FOR USING SETW'S
15 #include<string> // FOR USING STRINGS
16 #include<fstream>    // FOR USING OUTPUT FILES
17 #include<limits> // FOR USING LIMITS
18 #include<ios>        // FOR USING STREAM SIZE
19 #include<sstream>    // FOR USING OSTRINGSETREAM
20 #include"MyHeader.h"
21 using namespace std;
22
23
24 struct DVDNode
25 {
26     string      title;      //store Movie's title
27     string      leadActor;  //store leading Actor/Actress
28     string      supActor;   //store supporting Actor/Actress
29     string      genre;      //store Movie's genre
30     string      altGenre;   //store Movie's alternate genre
31     int         year;       //store Movie's year
32     int         rating;     //store Movie's rating
33     string      synopsis;   //store Movie's Synopsis
34     DVDNode     *next;      //linked list next pointer
35 };
36
37
38
39 class StackList
40 {
41 public:
42
43     //constructor
44     StackList();
45
46     //destructor
47     ~StackList();
48
49     /******************
50      **** MUTATORS ****
51      ******************/
52
53     void Push(DVDNode newDVD);  //create a DVDNode, add the
54                                 // node to the stack, by
55                                 // adding the node to front of the List
```

```
56
57      DVDNode Pop();              //returns the node from the top of the list
58                                  //removes the node from the list and delets it
59
60      /******************
61       **** ACCESSORS ***
62       ******************/
63
64      bool IsEmpty() const;       //checks if the list is empty
65
66      DVDNode Peek() const;       //returns the node at the top of the list
67
68      int Size() const;           //returns the number of the nodes in the list
69
70  protected:
71      DVDNode *head;              //head pointer for the list
72      int     stackCount;         //total number of nodes in the list
73
74  };
75
76
77
78  class MovieList: public StackList
79  {
80  public:
81
82      //constructor
83      MovieList();
84
85      //destructor
86      ~MovieList();
87
88      /******************
89       **** MUTATORS ****
90       ******************/
91
92      void CreatList(string inFileName);
93
94      /******************
95       **** ACCESSORS ***
96       ******************/
97
98      void OutputList(string outFileName) const;
99
100 private:
101
102     string WordWrap(string plot) const;
103
104     void PrintClassHeader(ostream &output,         //IN & OUT - utput file
105                           string  labName,         //OUT      - lab name
106                           int     labNumber,       //OUT  - lab number
107                           char    labType) const;  //OUT  - lab type
108
109
110
```

```
111 };
112
113
114
115
116
117 #endif /* CLASSHEADER_H_ */
118
```