

## Header.h

```
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
10 #ifndef HEADER_H_
11 #define HEADER_H_
12
13
14
15 #include<iostream> // INPUT AND OUTPUT
16 #include<iomanip>   // FOR USING SETW'S
17 #include<string>    // FOR USING STRINGS
18 #include<fstream>   // FOR USING OUTPUT FILES
19 #include<limits>    // FOR USING LIMITS
20 #include<ios>       // FOR USING STREAM SIZE
21 #include<sstream>   // FOR USING OSTRINGSTREAM
22 using namespace std;
23
24 /*****
25  * VARIABLES *
26  *****/
27
28 enum MenuOption
29 {
30     EXIT,
31     DISPLAY,
32     TITLE,
33     GENRE,
34     ACTOR,
35     YEAR,
36     RATE
37 };
38
39 struct MovieNode
40 {
41     string    title;
42     string    leadActor;
43     string    supportActor;
44     string    genre;
45     string    altGenre;
46     string    synopsis;
47     int       year;
48     int       rate;
49     MovieNode *next;
50     MovieNode *prev;
51 };
52
53 //FUNCTIONS
54
55 /*****/
```

## Header.h

```
56 * Function - PrintHeaderFile
57 * -----
58 * This function will output the class heading to the screen and output file
59 *
60 * return type - nothing
61 *           the function is void type
62 *****/
63 void PrintHeader (const string PROGRAMMER,
64                  const string CLASS,
65                  const string SECTION,
66                  const int    ASSIGN_NUM,
67                  const string ASSIGN_NAME,
68                  ofstream &outFile);
69
70
71 /*****
72 * Function - PrintMenu
73 * -----
74 * This function will output menu option to the screen and waits for the user to
75 * input an option to what to what to do. The options are as following:
76 *
77 * 1 - Output Entire List
78 * 2 - Title Search
79 * 3 - Genre Search
80 * 4 - Actor Search
81 * 5 - Year Search
82 * 6 - Rating Search (0-9)
83 * 0 - Exit
84 *
85 * return type - Integer
86 *****/
87 int PrintMenu();
88
89
90 /*****
91 * Function - CreatList
92 * -----
93 * This function will get the data from the input file and puts the data into
94 * the contents of the nodes, then add the nodes to the empty lists passed by
95 * Reference to the function
96 *
97 * return type - nothing
98 *           the function is void type
99 *****/
100 void CreatList(MovieNode *&head);
101
102
103 /*****
104 * Function - DisplayList
105 * -----
106 * This function will output the contents of the nodes of the linked list
107 * created in the CreatList function in the output file created by the user.
108 *
109 * return type - nothing
110 *           the function is void type
```

## Header.h

```
111 *****/
112 void DisplayList(MovieNode *head, ofstream &outFile);
113
114
115 *****/
116 * Function - SearchTitle
117 * -----
118 * This function will get a title from the user and search the title content of
119 * each node to see if there is a matching title in the nodes with the title
120 * searched by the user. if found, the function prints out every content of the
121 * node for the user, if not found, the function outputs that the title searched
122 * by the user was not found.
123 *
124 * return type - nothing
125 *             the function is void type
126 *****/
127 void SearchTitle(MovieNode *head, ofstream &outFile);
128
129
130 *****/
131 * Function - SearchGenre
132 * -----
133 * This function will get a genre from the user and search the genre content of
134 * each node to see if there is a matching genre in the nodes with the genre
135 * searched by the user. if found, the function prints out every content of
136 * every node for the user, if not found, the function outputs that the genre
137 * searched by the user was not found.
138 *
139 * return type - nothing
140 *             the function is void type
141 *****/
142 void SearchGenre(MovieNode *head, ofstream &outFile);
143
144
145 *****/
146 * Function - SearchActor
147 * -----
148 * This function will get a actor's name from the user and search the actors'
149 * content of each node to see if there is a matching actor's name in the nodes
150 * with the actor's name searched by the user. if found, the function prints out
151 * every content of every node for the user, if not found, the function outputs
152 * that the actor's name searched by the user was not found.
153 *
154 * return type - nothing
155 *             the function is void type
156 *****/
157 void SearchActor(MovieNode *head, ofstream &outFile);
158
159
160 *****/
161 * Function - SearchYear
162 * -----
163 * This function will get a year from the user and search the year content of
164 * each node to see if there is a matching year in the nodes with the year
165 * searched by the user. if found, the function prints out every content of
```

## Header.h

```
166 * every node for the user, if not found, the function outputs that the year
167 * searched by the user was not found.
168 *
169 * return type - nothing
170 *           the function is void type
171 *****/
172 void SearchYear(MovieNode *head, ofstream &outFile);
173
174
175 /*****
176 * Function - SearchRate
177 * -----
178 * This function will get a rating from the user and search the genre content of
179 * each node to see if there is a matching rating in the nodes with the rating
180 * searched by the user. if found, the function prints out every content of
181 * every node for the user, if not found, the function outputs that the rating
182 * searched by the user was not found.
183 *
184 * return type - nothing
185 *           the function is void type
186 *****/
187 void SearchRate(MovieNode *head, ofstream &outFile);
188
189
190 #endif /* HEADER_H_ */
191
```