

SearchActor.cpp

```
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 #include "Header.h"
11
12 /*****
13 * Function - SearchActor
14 * -----
15 * This function will get a actor's name from the user and search the actors'
16 * content of each node to see if there is a matching actor's name in the nodes
17 * with the actor's name searched by the user. if found, the function prints out
18 * every content of every node for the user, if not found, the function outputs
19 * that the actor's name searched by the user was not found.
20 *
21 * return type - nothing
22 *               the function is void type
23 *****/
24
25 void SearchActor(MovieNode *head, ofstream &outFile)
26 {
27     /*****
28      * VARIABLES *
29      *****/
30
31     MovieNode *movPtr; //Proc - stores the target name
32     MovieNode node;    //Proc & In - passes the info into persPtr
33
34
35     int i;              //Int - counts the founded data
36
37     /*****
38      * INITIALIZING *
39      *****/
40
41     i = 1;
42
43     /*****
44      * PROCESSING - searching for the actor in the list one by one
45      *****/
46
47
48     //if statement for empty list
49     if (head == NULL)
50     {
51         cout << endl;
52
53         cout << "Can not search for an actor in an empty list.";
54
55         cout << endl;
```

SearchActor.cpp

```

56     }
57
58     else
59     {
60         //creating new node
61
62         movPtr = new MovieNode;
63
64         //INPUT
65
66         cout << endl;
67
68         cout << "Which actor are you looking for? ";
69
70         getline(cin, node.leadActor);
71
72         *movPtr = node;
73
74         cout << endl;
75
76         cout << "Searching for the actor " << movPtr -> leadActor;
77
78         cout << endl;
79
80         outFile << "\nSearch by actor for " << movPtr -> leadActor
81             << " found:";
82
83         outFile << endl << endl;
84
85         outFile << left;
86
87         outFile << setw(9) << "MOVIE #" << " ";
88         outFile << setw(49) << "TITLE" << " ";
89         outFile << setw(6) << "YEAR" << " ";
90         outFile << setw(9) << "RATING" << " ";
91         outFile << setw(19) << "GENRE" << " ";
92         outFile << setw(19) << "ALT GENRE" << " ";
93         outFile << setw(20) << "LEAD ACTOR" << " ";
94         outFile << setw(20) << "SUPPORTING ACTOR" << " ";
95
96         outFile << left;
97
98         outFile << endl;
99
100        outFile << "----- ";
101        outFile << "----- ";
102        outFile << "----- ";
103        outFile << "----- ";
104        outFile << "----- ";
105        outFile << "----- ";
106        outFile << "----- ";
107        outFile << "----- ";
108        outFile << endl;
109
110        //GOES THROUGH THE LIST

```

```

111
112 while(head->next != NULL)
113 {
114     if ((head -> leadActor == movPtr -> leadActor) ||
115         (head -> supportActor == movPtr -> leadActor))
116     {
117
118         outFile << right;
119         outFile << setw(4) << i << "      " ;
120         outFile << left;
121         outFile << setw(49) << head -> title ;
122         outFile << setw(9) << head -> year ;
123         outFile << setw(9) << head -> rate ;
124         outFile << setw(19) << head -> genre ;
125         outFile << setw(19) << head -> altGenre ;
126         outFile << setw(20) << head -> leadActor ;
127         outFile << setw(20) << head -> supportActor;
128         outFile << endl;
129         outFile << left;
130
131         i++;
132         head = head -> next;
133
134     }
135
136     else
137     {
138         head = head -> next;
139     }
140
141 } // END - WHILE
142
143
144 if ((head -> next == NULL &&
145     (head -> leadActor == movPtr -> leadActor)) ||
146     (head -> next == NULL &&
147     (head -> supportActor == movPtr -> leadActor)))
148 {
149
150     outFile << right;
151     outFile << setw(4) << i << "      " ;
152     outFile << left;
153     outFile << setw(47) << head -> title ;
154     outFile << setw(8) << head -> year ;
155     outFile << setw(5) << head -> rate ;
156     outFile << setw(18) << head -> genre ;
157     outFile << setw(18) << head -> altGenre ;
158     outFile << setw(20) << head -> leadActor ;
159     outFile << setw(20) << head -> supportActor;
160     outFile << endl;
161     outFile << left;
162     i++;
163     head = head -> next;
164
165 }

```

SearchActor.cpp

```
166
167 //If couldn't found the name in the list
168 if (i == 1)
169 {
170     cout << "Sorry, no movies for the actor, " << movPtr -> genre
171         << " were found.";
172
173     cout << endl;
174 }
175
176 //Output the results in the output file
177 if(i > 1)
178 {
179     cout << "Found " << i-1 << " movies for the actor "
180         << movPtr -> genre << '!' << endl;
181 }
182
183 movPtr = NULL;
184 delete movPtr;
185
186 }
187
188 }
189
```