

## SearchYear.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 - SearchYear
14  * -----
15  * This function will get a year from the user and search the year content of
16  * each node to see if there is a matching year in the nodes with the year
17  * searched by the user. if found, the function prints out every content of
18  * every node for the user, if not found, the function outputs that the year
19  * searched by the user was not found.
20  *
21  * return type - nothing
22  *               the function is void type
23  *****/
24
25 void SearchYear(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     int checkInt;        //In - stores input value
38
39     bool checkInp;       //Proc - LCV value
40
41     /*****
42      * INITIALIZING *
43      *****/
44     checkInp = false;
45
46     i = 1;
47
48     /*****
49      * PROCESSING - searching for the year in the list one by one and output
50      *               each movie with the matching year to the output file
51      *****/
52
53
54     if (head == NULL)
55     {
```

## SearchYear.cpp

```
56     cout << endl;
57
58     cout << "Can not search for a year in an empty list.";
59
60     cout << endl;
61 }
62
63 else
64 {
65     //creating new node
66
67     movPtr = new MovieNode;
68
69     //INPUT
70
71     cout << endl;
72
73     do
74     {
75         //INPUT
76
77         cout << "\nWhich year are you looking for? ";
78
79         //CHECKS FOR THE CHAR INPUT
80
81         if (!(cin >> checkInt))
82         {
83
84             cin.clear();
85             cin.ignore(numeric_limits<streamsize>::max(), '\n');
86
87             cout << endl;
88             cout << "**** Please input a NUMBER between 1878 and 3000 ****";
89             cout << endl;
90             checkInp = false;
91
92         }
93
94         //CHECKS FOR THE RANGE ERROR
95
96
97
98
99         else if (checkInt >= 3001 || checkInt <= 1877 )
100         {
101
102             cout << endl;
103             cout << "**** The number " << checkInt
104                 << " is an invalid entry ****" << endl;
105             cout << "**** Please input a number between 1878 and 3000 ****";
106             cout << endl;
107
108             checkInp = false;
109
110         }
```

# SearchYear.cpp

```

111
112 //PASS
113
114 else
115 {
116
117     cin.ignore(numeric_limits<streamsize>::max(), '\n');
118     checkInp = true;
119
120 }
121
122 }while(!checkInp);
123
124 node.year = checkInt;
125
126 *movPtr = node;
127
128 cout << endl;
129
130 cout << "Searching for the year " << movPtr -> year;
131
132 cout << endl;
133
134 outFile << "\nSearch by actor for " << movPtr -> year << " found:";
135
136 outFile << endl << endl;
137
138 outFile << left;
139
140 outFile << setw(9) << "MOVIE #" << " ";
141 outFile << setw(49) << "TITLE" << " ";
142 outFile << setw(6) << "YEAR" << " ";
143 outFile << setw(9) << "RATING" << " ";
144 outFile << setw(19) << "GENRE" << " ";
145 outFile << setw(19) << "ALT GENRE" << " ";
146 outFile << setw(20) << "LEAD ACTOR" << " ";
147 outFile << setw(20) << "SUPPORTING ACTOR" << " ";
148
149 outFile << left;
150
151 outFile << endl;
152
153 outFile << "----- ";
154 outFile << "----- ";
155 outFile << "---- ";
156 outFile << "----- ";
157 outFile << "----- ";
158 outFile << "----- ";
159 outFile << "----- ";
160 outFile << "----- ";
161 outFile << endl;
162
163 //GOES THROUGH THE LIST
164
165 while(head->next != NULL)

```

# SearchYear.cpp

```

166 {
167     if (head -> year == movPtr -> year)
168     {
169
170         outFile << right;
171         outFile << setw(4) << i << "      " ;
172         outFile << left;
173         outFile << setw(47) << head -> title ;
174         outFile << setw(8) << head -> year ;
175         outFile << setw(5) << head -> rate ;
176         outFile << setw(18) << head -> genre ;
177         outFile << setw(18) << head -> altGenre ;
178         outFile << setw(20) << head -> leadActor ;
179         outFile << setw(20) << head -> supportActor;
180         outFile << endl;
181         outFile << left;
182
183         i++;
184         head = head -> next;
185
186     }
187
188     else
189     {
190         head = head -> next;
191     }
192
193 } // END - WHILE
194
195
196 if (head -> next == NULL && (head -> year == movPtr -> year))
197 {
198
199     outFile << right;
200     outFile << setw(4) << i << "      " ;
201     outFile << left;
202     outFile << setw(47) << head -> title ;
203     outFile << setw(8) << head -> year ;
204     outFile << setw(5) << head -> rate ;
205     outFile << setw(18) << head -> genre ;
206     outFile << setw(18) << head -> altGenre ;
207     outFile << setw(20) << head -> leadActor ;
208     outFile << setw(20) << head -> supportActor;
209     outFile << endl;
210     outFile << left;
211     i++;
212     head = head -> next;
213
214 }
215
216 //IF NOT FOUND IN THE LIST
217
218 if (i == 1)
219 {
220     cout << endl;

```

## SearchYear.cpp

```
221
222     cout << "Sorry, no movies for the year, " << movPtr -> year
223         << " were found.";
224
225     cout << endl;
226 }
227
228 //OUTPUT RESULT INTO THE CONSOLE
229
230 if(i > 1)
231 {
232     cout << "Found " << i-1 << " movies for the year "
233         << movPtr -> year << '!' << endl;
234 }
235
236 movPtr = NULL;
237 delete movPtr;
238
239 }
240
241 }
242
```