```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * Assign #3  : Searching linked list
 * DUE DATE   : 6 November 2019
 *****************************************************************************/

#include "Header.h"

/*****************************************************************************
 * Function - SearchTitle
 * ---------------------------------------------------------------------------
 * This function will get a title from the user and search the title content of
 * each node to see if there is a matching title in the nodes with the title
 * searched by the user. if found, the function prints out every content of the
 * node for the user, if not found, the function outputs that the title searched
 * by the user was not found.
 *
 * return type - nothing
 *               the function is void type
 *****************************************************************************/

void SearchTitle(MovieNode *head, ofstream &outFile)
{
    /*************
     * VARIABLES *
     *************/

    //NODES

    MovieNode *movPtr;      //PROCESS - stores the target name

    MovieNode node;             //PROCESS & IN - passes the info into persPtr


    //BOOLS

    bool found;             //PROCESS - condition value for the searched name


    //INTEGERS

    int stringLength;

    int maxLength;

    int lineLen;

    int wordLen;

    int i;


```

```
56      //STRINGS
57
58      string word;
59
60      string line;
61
62      string plot;
63
64      /****************
65       * INITIALIZING *
66       ****************/
67
68      word.clear();
69      line.clear();
70
71      maxLength = 75;
72
73      /*****************************************************************************
74       * OUTPUT - outputs the title of the movie, the lead actor and the support
75       *          actor, genre and the alternative genre and the plot of the movie
76       *****************************************************************************/
77
78
79      //CHECKS FOR EMPTY LIST
80
81      found = false;
82
83      if (head == NULL)
84      {
85          cout << endl;
86
87          cout << "Can not search for the title in an empty list.";
88
89          cout << endl;
90      }
91
92      else
93      {
94          //creating new node
95
96          movPtr = new MovieNode;
97
98          //INPUT
99
100         cout << endl;
101
102         cout << "Which title are you looking for? ";
103         getline(cin, node.title);
104
105         *movPtr = node;
106
107         cout << endl << endl;
108
109         cout << "Searching for the title " << movPtr -> title;
110
```

```
111         cout << endl;
112
113         //GOES ATHROUGH THE LIST
114
115         while ((head -> next != NULL) && !found)
116         {
117             //IF FOUND
118
119             if (head -> title == movPtr -> title)
120             {
121                 //OUTPUT
122
123                 cout << "Found the movie " << head -> title << "!" << endl;
124
125                 outFile << endl;
126
127                 outFile << left;
128                 outFile << "**************************************************"
129                       << "**************************************************"
130                       << endl;
131
132                 outFile << "Title: " << head -> title << endl;
133
134                 outFile << "--------------------------------------------------"
135                       << "--------------------------------------------------"
136                       << endl;
137
138                 outFile << setw(6)   << "Year:"   << setw (12) << head -> year;
139                 outFile << setw(8)   << "Rating:" << head -> rate << endl;
140
141                 outFile << "--------------------------------------------------"
142                       << "--------------------------------------------------"
143                       << endl;
144
145                 outFile << setw(18) << "Leading Actor:"
146                       << setw(25) << head -> leadActor;
147
148                 outFile << setw(9)  << "Genre 1:" << head -> genre << endl;
149
150
151                 outFile << setw(18) << "Supporting Actor:"
152                       << setw(25) << head -> supportActor;
153
154                 outFile << setw(9)  << "Genre 2:" << head -> altGenre << endl;
155
156                 outFile << "--------------------------------------------------"
157                       << "--------------------------------------------------"
158                       << endl;
159
160                 outFile << "PLOT:" << endl;
161
162                 stringLength  = head -> synopsis.length();
163
164                 cout << head -> synopsis;
165
```

```
166                 plot = head -> synopsis;
167
168                 for(i = 0; i <= stringLength; i++)
169                 {
170                      if(plot[i] != ' ')
171                      {
172                          word = word + head -> synopsis[i];
173                      }
174
175                      else
176                      {
177                          lineLen = line.length();
178                          wordLen = word.length();
179
180                          if((lineLen + wordLen) > maxLength)
181                          {
182                              outFile << plot;
183
184                              line.clear();
185
186                          }
187
188                          line = word + " ";
189
190                          word.clear();
191                      }
192
193                 } //END - FOR
194
195                 outFile << "*****************************************************"
196                       << "*****************************************"
197                       << endl << endl;
198
199                 found = true;
200
201             } //END - IF
202
203             else
204             {
205
206                 head  = head -> next;
207
208                 found = false;
209
210             }
211
212
213         } //END - WHILE
214
215         if (head -> next == NULL && head -> title == movPtr -> title)
216         {
217             //OUTPUT
218
219             outFile << endl;
220
```

```
221             outFile << left;
222             outFile << "*****************************************************"
223                   << "*****************************************************"
224                   << endl;
225
226             outFile << "Title: " << head -> title << endl;
227
228             outFile << "-----------------------------------------------------"
229                   << "-----------------------------------------------------"
230                   << endl;
231
232             outFile << setw(6)   << "Year:"   << setw (12) << head -> year;
233             outFile << setw(8)   << "Rating:" << head -> rate << endl;
234
235             outFile << "-----------------------------------------------------"
236                   << "-----------------------------------------------------"
237                   << endl;
238
239             outFile << setw(18) << "Leading Actor:"
240                   << setw(25) << head -> leadActor;
241
242             outFile << setw(9)  << "Genre 1:" << head -> genre << endl;
243
244
245             outFile << setw(18) << "Supporting Actor:"
246                   << setw(25) << head -> supportActor;
247
248             outFile << setw(9)  << "Genre 2:" << head -> altGenre << endl;
249
250             outFile << "-----------------------------------------------------"
251                   << "-----------------------------------------------------"
252                   << endl;
253
254             outFile << "PLOT:" << endl;
255
256             stringLength  = head -> synopsis.length();
257
258             plot = head -> synopsis;
259
260             for(i = 0; i <= stringLength; i++)
261             {
262                  if(plot[i] != ' ')
263                  {
264
265                      word = word + head -> synopsis[i];
266
267                  }
268
269                  else
270                  {
271                      lineLen = line.length();
272                      wordLen = word.length();
273
274                      if((lineLen + wordLen) > maxLength)
275                      {
```

```
276                         outFile << line;
277                         line.clear();
278
279                     }
280
281                     line = word + " ";
282                     word.clear();
283                 }
284
285             } // END - FOR
286
287             outFile << "*********************************************************"
288                   << "*********************************************************"
289                   << endl << endl;
290
291             found = true;
292
293         } //END - IF
294
295         //IF NOT FOUND IN THE LIST
296
297         else if (!found)
298         {
299             cout << "Sorry, the movie \"" << movPtr -> title;
300             cout << "\" was not found.";
301
302             cout << endl;
303         }
304
305     }
306
307    //JUST TOBE SAFE
308     movPtr = NULL;
309 }
310
```