```
 1 ************************************************************
 2 *   PROGRAMMED BY : Andrew Gharios
 3 *   STUDENT ID    : 1449366
 4 *   CLASS         : M-Th 5-7:20p
 5 *   ASSIGNMENT #5 : OPP - DVD Movie List
 6 ************************************************************
 7 ***************************************************************************
 8 Title: Antwone Fisher
 9 ---------------------------------------------------------------------------
10 Year: 2002        Rating: 7
11 ---------------------------------------------------------------------------
12 Leading Actor:    Denzel Washington      Genre 1: Biography
13 Supporting Actor: Derek Luke             Genre 2: Drama
14 ---------------------------------------------------------------------------
15 PLOT:
16 Antwone Fisher, a young navy man, is forced to see a psychiatrist after a
17 violent outburst against a fellow crewman. During the course of treatment a
18 painful past is revealed and a new hope begins.
19 ***************************************************************************
20
21
22 ***************************************************************************
23 Title: Bagger Vance, The Legend of
24 ---------------------------------------------------------------------------
25 Year: 2000        Rating: 6
26 ---------------------------------------------------------------------------
27 Leading Actor:    Will Smith             Genre 1: Drama
28 Supporting Actor: Matt Damon             Genre 2: Drama
29 ---------------------------------------------------------------------------
30 PLOT:
31 A down-and-out golfer attempts to recover his game and his life with help
32 from a mystical caddy
33 ***************************************************************************
34
35
36 ***************************************************************************
37 Title: Cinderella Man
38 ---------------------------------------------------------------------------
39 Year: 2005        Rating: 8
40 ---------------------------------------------------------------------------
41 Leading Actor:    Russell Crowe          Genre 1: Biography
42 Supporting Actor: Renee Zellweger        Genre 2: Drama
43 ---------------------------------------------------------------------------
44 PLOT:
45 The story of James Braddock, a supposedly washed up boxer who came back to
46 become a champion and an inspiration in the 1930s.
47 ***************************************************************************
48
49
```

```
**************************************************************************
Title: Five People You Meet in Heaven, The
--------------------------------------------------------------------------
Year: 2004        Rating: 7
--------------------------------------------------------------------------
Leading Actor:    Jon Voight            Genre 1: Drama
Supporting Actor: Ellen Burstyn         Genre 2: Drama
--------------------------------------------------------------------------
PLOT:
On his 83rd birthday, Eddie (Voight), a war vet and a maintenance worker at
the Ruby Pier amusement park, dies while trying to save a girl who is
sitting under a falling ride. When he awakens in the afterlife, he
encounters five people with ties to his corporeal existence who help him
understand the meaning of his life.
**************************************************************************


**************************************************************************
Title: Good Night, Good Luck
--------------------------------------------------------------------------
Year: 2005        Rating: 8
--------------------------------------------------------------------------
Leading Actor:    David Strathairn      Genre 1: Biography
Supporting Actor: George Clooney        Genre 2: Drama
--------------------------------------------------------------------------
PLOT:
In the early 1950's, the threat of Communism created an air of paranoia in
the United States and exploiting those fears was Senator Joseph McCarthy of
Wisconsin. However, CBS reporter Edward R. Murrow and his producer Fred W.
Friendly decided to take a stand and challenge McCarthy and expose him for
the fear monger he was. However, their actions took a great personal toll
on both men, but they stood by their convictions and helped to bring down
one of the most controversial senators in American history.
**************************************************************************


**************************************************************************
Title: Invictus
--------------------------------------------------------------------------
Year: 2009        Rating: 8
--------------------------------------------------------------------------
Leading Actor:    Morgan Freeman        Genre 1: Biography
Supporting Actor: Matt Damon            Genre 2: Drama
--------------------------------------------------------------------------
PLOT:
Nelson Mandela, in his first term as the South African President, initiates
a unique venture to unite the apartheid-torn land: enlist the national
rugby team on a mission to win the 1995 Rugby World Cup.
**************************************************************************
```

```
 99
100
101 ********************************************************************************
102 Title: Miss Petigrew Lives for a Day
103 --------------------------------------------------------------------------------
104 Year: 2008        Rating: 7
105 --------------------------------------------------------------------------------
106 Leading Actor:    Frances McDomand       Genre 1: Comedy
107 Supporting Actor: Amy Adams              Genre 2: Romance
108 --------------------------------------------------------------------------------
109 PLOT:
110 Guinevere Pettigrew, a middle-aged London governess, finds herself unfairly
111 dismissed from her job. An attempt to gain new employment catapults her
112 into the glamorous world and dizzying social whirl of an American actress
113 and singer, Delysia Lafosse.
114 ********************************************************************************
115
116
117 ********************************************************************************
118 Title: Pay it Forward
119 --------------------------------------------------------------------------------
120 Year: 2000        Rating: 7
121 --------------------------------------------------------------------------------
122 Leading Actor:    Kevin Spacey           Genre 1: Drama
123 Supporting Actor: Helen Hunt             Genre 2: Romance
124 --------------------------------------------------------------------------------
125 PLOT:
126 Like some other kids, 12-year-old Trevor McKinney believed in the goodness
127 of human nature. Like many other kids, he was determined to change the
128 world for the better. Unlike most other kids, he succeeded.
129 ********************************************************************************
130
131
132 ********************************************************************************
133 Title: Pianist, The
134 --------------------------------------------------------------------------------
135 Year: 2003        Rating: 9
136 --------------------------------------------------------------------------------
137 Leading Actor:    Adrien Brody           Genre 1: Biography
138 Supporting Actor: Thomas Kreschmann      Genre 2: Drama
139 --------------------------------------------------------------------------------
140 PLOT:
141 A Polish Jewish musician struggles to survive the destruction of the Warsaw
142 ghetto of World War II.
143 ********************************************************************************
144
145
146 ********************************************************************************
147 Title: Pirates of the Caribbean 1: The Curse of the Bl...
```

```
148 ---------------------------------------------------------------------------
149 Year: 2003        Rating: 8
150 ---------------------------------------------------------------------------
151 Leading Actor:    Johnny Depp            Genre 1: Action
152 Supporting Actor: Orlando Bloom          Genre 2: Action
153 ---------------------------------------------------------------------------
154 PLOT:
155 This swash-buckling tale follows the quest of Captain Jack Sparrow, a savvy
156 pirate, and Will Turner, a resourceful blacksmith, as they search for
157 Elizabeth Swann. Elizabeth, the daughter of the governor and the love of
158 Will's life, has been kidnapped by the feared Captain Barbossa. Little do
159 they know, but the fierce and clever Barbossa has been cursed. He, along
160 with his large crew, is under an ancient curse, doomed for eternity to
161 neither live, nor die. That is, unless a blood sacrifice is made. Full of
162 edge-of-the-seat action and swashbuckling adventures, this is a movie you
163 won't want to miss!
164 ***************************************************************************
165
166
167 ***************************************************************************
168 Title: Pursuit of Happyness
169 ---------------------------------------------------------------------------
170 Year: 2008        Rating: 8
171 ---------------------------------------------------------------------------
172 Leading Actor:    Will Smith             Genre 1: Biography
173 Supporting Actor: Jaden Smith            Genre 2: Drama
174 ---------------------------------------------------------------------------
175 PLOT:
176 A struggling salesman takes custody of his son as he's poised to begin a
177 life-changing professional endeavor.
178 ***************************************************************************
179
180
181 ***************************************************************************
182 Title: Rent
183 ---------------------------------------------------------------------------
184 Year: 2005        Rating: 7
185 ---------------------------------------------------------------------------
186 Leading Actor:    Anthony Rapp           Genre 1: Musical
187 Supporting Actor: Adam Pascal            Genre 2: Drama
188 ---------------------------------------------------------------------------
189 PLOT:
190 This is the film version of the Pulitzer and Tony Award winning musical
191 about Bohemians in the East Village of New York City struggling with life,
192 love and AIDS, and the impacts they have on America.
193 ***************************************************************************
194
195
196 ***************************************************************************
```

```
Title: Secondhand Lions
--------------------------------------------------------------------------
Year: 2003        Rating: 8
--------------------------------------------------------------------------
Leading Actor:    Haley Joel Osmet       Genre 1: Comedy
Supporting Actor: Robert Duvall          Genre 2: Comedy
--------------------------------------------------------------------------
PLOT:
A coming-of-age story about a shy, young boy sent by his irresponsible
mother to spend the summer with his wealthy, eccentric uncles in Texas.
**************************************************************************


**************************************************************************
Title: Shackelton
--------------------------------------------------------------------------
Year: 2002        Rating: 8
--------------------------------------------------------------------------
Leading Actor:    Kenneth Branagh        Genre 1: Biography
Supporting Actor: John Grillo            Genre 2: Drama
--------------------------------------------------------------------------
PLOT:
The true story of Shackleton's 1914 Endurance expedition to the the South
Pole and his epic struggle to lead his 28 man crew to safety after his ship
was crushed in the pack ice.
**************************************************************************


**************************************************************************
Title: Shawshank Redemption, The
--------------------------------------------------------------------------
Year: 1994        Rating: 9
--------------------------------------------------------------------------
Leading Actor:    Tim Robbins            Genre 1: Drama
Supporting Actor: Morgan Freeman         Genre 2: Crime
--------------------------------------------------------------------------
PLOT:
Two imprisoned men bond over a number of years, finding solace and eventual
redemption through acts of common decency.
**************************************************************************


**************************************************************************
Title: Thelma and Louise
--------------------------------------------------------------------------
Year: 1991        Rating: 7
--------------------------------------------------------------------------
Leading Actor:    Susan Sarandon         Genre 1: Action
Supporting Actor: Geena Davis            Genre 2: Action
```

```
246 ---------------------------------------------------------------------------
247 PLOT:
248 Whilst on a short weekend getaway, Louise shoots a man who had tried to
249 rape Thelma. Due to the incriminating circumstances, they make a run for it
250 but are soon followed closely by the authorities including a local
251 policeman who is sympathetic to their plight. The federal authorities,
252 however, have less compassion and thus a cross country chase ensues for the
253 two fugitives. Along the way, both women rediscover the strength of their
254 friendship and surprising aspects of their personalities and self-strengths
255 in the trying times.
256 ***************************************************************************
257
258
259 ***************************************************************************
260 Title: Unfinished Life, An
261 ---------------------------------------------------------------------------
262 Year: 2005        Rating: 7
263 ---------------------------------------------------------------------------
264 Leading Actor:    Robert Redford         Genre 1: Drama
265 Supporting Actor: Jennifer Lopez         Genre 2: Drama
266 ---------------------------------------------------------------------------
267 PLOT:
268 A down on her luck woman, desperate to provide care for her daughter, moves
269 in with her father in-law from whom she is estranged. Through time, they
270 learn to forgive each other and heal old wounds.
271 ***************************************************************************
272
273
274 ***************************************************************************
275 Title: Whip It
276 ---------------------------------------------------------------------------
277 Year: 2009        Rating: 7
278 ---------------------------------------------------------------------------
279 Leading Actor:    Ellen Page             Genre 1: Drama
280 Supporting Actor: Marcia Gay Harden      Genre 2: Sport
281 ---------------------------------------------------------------------------
282 PLOT:
283 In Bodeen, Texas, an indie-rock loving misfit finds a way of dealing with
284 her small-town misery after she discovers a roller derby league in nearby
285 Austin.
286 ***************************************************************************
287
288
289 ***************************************************************************
290 Title: X-Men
291 ---------------------------------------------------------------------------
292 Year: 2000        Rating: 7
293 ---------------------------------------------------------------------------
294 Leading Actor:    Hugh Jackman           Genre 1: Action
```

```
Supporting Actor: Patrick Stewart        Genre 2: Action
--------------------------------------------------------------------------------
PLOT:
All over the planet, unusual children are born with an added twist to their
genetic code. This X-factor allows the children to perform extraordinary
feats - flight, telekinetics, laser beams from the eyes and more. One Dr.
Charles Xavier gathers the children to a place where he can train them to
use their powers for themselves and the forces of good he dubs these
children his X-men and hijinks ensue.
********************************************************************************


********************************************************************************
Title: Yankee Doodle Dandee
--------------------------------------------------------------------------------
Year: 1942        Rating: 8
--------------------------------------------------------------------------------
Leading Actor:    James Cagney           Genre 1: Musical
Supporting Actor: Joan Leslie            Genre 2: Biography
--------------------------------------------------------------------------------
PLOT:
This film depicts the life of the renowned musical composer, playwright,
actor, dancer and singer George M. Cohan.
********************************************************************************


```

```
1  ************************************************************
2  *   PROGRAMMED BY : Andrew Gharios
3  *   STUDENT ID    : 1449366
4  *   CLASS         : M-Th 5-7:20p
5  *   ASSIGNMENT #5 : OPP - DVD Movie List
6  ************************************************************
7  Which input file would you like to use? inFile.txt
8  Which output file would you like to use? OutFile.txt
9
10 Name: Yankee Doodle Dandee
11 Name: X-Men
12 Name: Whip It
13 Name: Unfinished Life, An
14 Name: Thelma and Louise
15 Name: Shawshank Redemption, The
16 Name: Shackelton
17 Name: Secondhand Lions
18 Name: Rent
19 Name: Pursuit of Happyness
20 Name: Pirates of the Caribbean 1: The Curse of the Black Pearl
21 Name: Pianist, The
22 Name: Pay it Forward
23 Name: Miss Petigrew Lives for a Day
24 Name: Invictus
25 Name: Good Night, Good Luck
26 Name: Five People You Meet in Heaven, The
27 Name: Cinderella Man
28 Name: Bagger Vance, The Legend of
29 Name: Antwone Fisher
30
31 Writing output file OutFile.txt
```

```
 1  #ifndef HEADER_H_
 2  #define HEADER_H_
 3
 4  #include <iostream> // cin, cout.
 5  #include <string>   // string datatype variables.
 6  #include <fstream>  // Fstream files.
 7  #include <iomanip>  // fixed, setw, setprecision.
 8  #include <ostream>
 9  #include <sstream>
10  using namespace std;
11
12  enum Menu {
13      EXIT = 0,
14      OutputList,
15      TitleSearch,
16      GenreSearch,
17      ActorSearch,
18      YearSearch,
19      RatingSearch,
20  };
21
22  struct DVDNode
23  {
24      string title;
25      string leadAct;
26      string supAct;
27      string genre;
28      string altGenre;
29      int    year;
30      int    rating;
31      string synop;
32      DVDNode* next;
33  };
34
35  const int ColSpace = 18; // CALC  - Column setw space size.
36
37  #endif
38
39
```

```cpp
#include "Header.h"
#include "Movie.h"
#include "Stack.h"

/*****************************************************************************
**
* OPP - DVD Movie List.
*----------------------------------------------------------------------------
* This program will keep track of all the Dvds for the user and display them
* into an output file.
*----------------------------------------------------------------------------
* INPUT:
*  userInFile : Input file name.
*  userOFile  : Output file name.
*****************************************************************************
/
int main()
{
    /*************************************************************************
    **
    * CONSTANTS
    * ------------------------------------------------------------------------
    * OUTPUT - USED FOR CLASS HEADING
    * ------------------------------------------------------------------------
    * PROGRAMMER : Programmer's Name
    * CLASS      : Student's Course
    * SECTION    : Class Days and Times
    * LAB_NUM    : Lab Number (specific to this lab)
    * LAB_NAME   : Title of the Lab
    *************************************************************************/

    const string AS_NAME = "OPP - DVD Movie List";
    const int AS_NUM = 5;
    const string STUDENT_NAME = "Andrew Gharios";
    const string CLASS_INFO = "M-Th 5-7:20p";
    const char AS_TYPE = 'A';
    const long long STUDENT_ID = 1449366;

    MovieList movie;    // CALC       - Movies list.
    string userInFile;  // IN & CALC  - User input file.
    string userOFile;   // IN & CALC  - User output file.

    cout << left;
    cout << "*****************************************************************\n";
    cout << "*   PROGRAMMED BY : " << STUDENT_NAME << endl;
    cout << "*   " << setw(14) << "STUDENT ID " << ": " << STUDENT_ID << endl;
    cout << "*   " << setw(14) << "CLASS " << ": " << CLASS_INFO << endl;
```

```
45      cout << "*    ";
46
47      // PROCESSING - This will adjust setws and format appropriately based
48      //              on if this is a lab 'L' or assignment
49
50      if (toupper(AS_TYPE) == 'L')
51      {
52          cout << "LAB #" << setw(9);
53      }
54      else
55      {
56          cout << "ASSIGNMENT #" << setw(2);
57      }
58      cout << AS_NUM << ": " << AS_NAME << endl;
59      cout << "************************************************************";
60      cout << right << endl;
61
62
63      cout << "Which input file would you like to use? ";
64      getline(cin, userInFile);
65      cout << "Which output file would you like to use? ";
66      getline(cin, userOFile);
67      cout << endl;
68
69      movie.CreateList(userInFile);
70      movie.DisplayList(userOFile);
71
72      return 0;
73  }
74
75
76
77
78
```

```
1  #ifndef STACK_H_
2  #define STACK_H_
3
4  #include "Header.h"
5
6  class StackList
7  {
8  public:
9
10     StackList();  // CONSTRUCTOR
11     ~StackList(); // DECONSTRUCTOR
12
13     /*********************
14      **    MUTATORS    **
15      *********************/
16     void Push(DVDNode newDVD); // Adds a new Node to the list.
17     DVDNode Pop();  // Removes first node and returns it.
18
19
20     /*********************
21      **    ACCESSORS    **
22      *********************/
23     bool isEmpty() const; // Returns if list is empty.
24     DVDNode Peek() const; // Returns the first object of the list.
25     int Size() const;     // Returns the size of the list.
26
27 protected:
28     DVDNode* head;       // CALC - Stack head pointer.
29     int      stackCount; // CALC - Amount of objects in stack.
30 };
31
32 #endif
```

```cpp
#include "Header.h"
#include "Stack.h"

StackList::StackList() // CONSTRUCTOR
{
    head = NULL;
    stackCount = 0;
}

StackList::~StackList() // DECONSTRUCTOR
{
    DVDNode* dvdPtr; // CALC - Dvd pointer.

    dvdPtr = head;

    while (dvdPtr != NULL)
    {
        head = head->next;
        delete dvdPtr;
        dvdPtr = head;
    }
}

void StackList::Push(DVDNode newDVD)
{
    DVDNode* dvdPtr; // CALC - Dvd pointer.
    dvdPtr = new DVDNode;

    *dvdPtr = newDVD;

    dvdPtr->next = head;
    head = dvdPtr;
}

DVDNode StackList::Pop()
{
    DVDNode* dvdPtr; // CALC - Dvd pointer.

    dvdPtr = head;
    head = dvdPtr->next;
    delete dvdPtr;
    dvdPtr = NULL;

    return *head;
}

bool StackList::isEmpty() const
{
    if (head == NULL)
```

```
50      {
51          return true;
52      }
53      else
54      {
55          return false;
56      }
57  }
58
59  DVDNode StackList::Peek() const
60  {
61      return *head;
62  }
63
64  int StackList::Size() const
65  {
66      DVDNode* ptr; // CALC - DVDpointer.
67      int counter;  // CALC - List counter.
68
69      counter = 0;
70
71      while (ptr != NULL)
72      {
73          counter++;
74          ptr = ptr->next;
75      }
76      return counter;
77  }
78
79  /*****************************************************************************
    **
80  * Push
81  *   This method will create a node and add it to the front.
82  * INPUTS:
83  *   newDVD : new DVD node.
84  *
85  * No outputs.
86  *****************************************************************************
    */
87
88  /*****************************************************************************
    **
89  * Pop
90  *   This method will remove the first object and return it.
91  * No inputs.
92  *
```

```
 93  * OUTPUTS:
 94  *  head : removed Node.
 95  *                                                                        ↵
       ************************************************************************ ↵
       */
 96
 97  /                                                                       ↵
       ************************************************************************ ↵
       **
 98  * IsEmpty
 99  *   This method will check if the list is empty and return a boolean     ↵
       expression.
100  * No inputs.
101  *
102  * OUTPUTS:
103  *  bool : If list is empty or not.
104  *                                                                        ↵
       ************************************************************************ ↵
       */
105
106 /                                                                       ↵
       ************************************************************************ ↵
       **
107 * Peek
108 *   This method will check the first node of the list and return a copy of it.
109 * No inputs.
110 *
111 * OUTPUTS:
112 *  head : First Node of the list.
113 *                                                                        ↵
       ************************************************************************ ↵
       */
114
115 /                                                                       ↵
       ************************************************************************ ↵
       **
116 * Size
117 *   This method will count the size of the list.
118 * No inputs.
119 *
120 * OUTPUTS:
121 *  counter : List size.
122 *                                                                        ↵
       ************************************************************************ ↵
       */
```

```
1  #ifndef MOVIE_H_
2  #define MOVIE_H_
3
4  #include "Header.h"
5  #include "Stack.h"
6
7  class MovieList : public StackList
8  {
9  public:
10     MovieList();  // DECONSTRUCTOR
11     ~MovieList(); // CONSTRUCTOR
12
13     /*********************
14      **    MUTATORS    **
15      *********************/
16     void CreateList(string IFileName);             // Creates a lit of DVDs.
17     void DisplayList(string outputFileName) const; // Displays the list of
         DVDs.
18
19 private:
20     /*********************
21      **    ACCESSORS    **
22      *********************/
23     string WordWrap(string plot) const; // returns plot limited to characters
         per line.
24     string TitleString(string title) const;   // returns the Title without
         exceeding limit.
25     void   PrintHeader(ostream& oFile) const; // Outputs the class header to
         Ofile.
26 };
27
28 #endif
```

```cpp
#include "Header.h"
#include "Movie.h"

MovieList::MovieList(){}
MovieList::~MovieList(){}

void MovieList::CreateList(string inFileName)
{
    DVDNode* dvdPtr; // CALC - DVD pointer.
    dvdPtr = new DVDNode;

    ifstream inFile;
    inFile.open(inFileName.c_str());

    while (inFile)
    {
        getline(inFile, dvdPtr->title);
        getline(inFile, dvdPtr->leadAct);
        getline(inFile, dvdPtr->supAct);
        getline(inFile, dvdPtr->genre);
        getline(inFile, dvdPtr->altGenre);
        inFile >> dvdPtr->year;
        inFile.ignore(10000, '\n');
        inFile >> dvdPtr->rating;
        inFile.ignore(10000, '\n');
        getline(inFile, dvdPtr->synop);
        inFile.ignore(10000, '\n');

        cout << "Name: " << dvdPtr->title << endl;

        Push(*dvdPtr);
    }
    delete dvdPtr;
    dvdPtr = NULL;
    inFile.close();
}

void MovieList::DisplayList(string outputFileName) const
{
    DVDNode* dvdPtr; // CALC   - DVD pointer
    int counter; // CALC   - Counter
    ofstream oFile;  // CALC & OUT - Output file variable.

    counter = 1;
    dvdPtr = head;

    oFile.open(outputFileName.c_str());

    PrintHeader(oFile);
```

```
50
51      cout << "\nWriting output file " << outputFileName << endl;
52
53      while (dvdPtr != NULL)
54      {
55          oFile << left;
56          oFile << "**********************************************************************
            ********" << endl;
57          oFile << "Title: " << TitleString(dvdPtr->title) << endl;
58          oFile << "Title: " << TitleString(dvdPtr->title) << endl;
59          oFile << "----------------------------------------------------------------------
            --------" << endl;
60          oFile << "Year: " << setw(ColSpace - 6) << dvdPtr->year << "Rating: "
              << dvdPtr->rating << endl;
61          oFile << "----------------------------------------------------------------------
            --------" << endl;
62          oFile << setw(ColSpace) << "Leading Actor: " << setw(ColSpace + 5) <<
              dvdPtr->leadAct << "Genre 1: ";
63          oFile << dvdPtr->genre << endl;
64          oFile << setw(ColSpace) << "Supporting Actor: " << setw(ColSpace + 5)
              << dvdPtr->supAct << "Genre 2: ";
65          oFile << dvdPtr->altGenre << endl;
66          oFile << "----------------------------------------------------------------------
            --------" << endl;
67          oFile << "PLOT:\n";
68          oFile << WordWrap((*dvdPtr).synop);
69          oFile << "\n**********************************************************************
            **********" << endl;
70          oFile << endl << endl;
71          oFile << right;
72          counter++;
73          dvdPtr = dvdPtr->next;
74      }
75      oFile.close();
76  }
77
78  string MovieList::WordWrap(string title) const
79  {
80      const int MAX = 75; // CALC - Maximum characters per line.
81
82      ostringstream stream; // Stringstream output variable
83      string word; // CALC & OUT - Temp word memory.
84      string line; // CALC & OUT - Temp line memory.
85
```

```
 86      word = "";
 87      line = "";
 88
 89      for (unsigned int i = 0; i < title.length(); i++)
 90      {
 91          if (title[i] != ' ')
 92          {
 93              word += title[i];
 94          }
 95          else
 96          {
 97              if (line.length() + word.length() > MAX)
 98              {
 99                  stream << line << endl;
100                  line = "";
101              }
102              line += word + " ";
103              word.clear();
104          }
105          if (i == title.length() - 1)
106          {
107              if (line.length() + word.length() > MAX)
108              {
109                  stream << line << endl;
110                  line.clear();
111              }
112              line += word + " ";
113              stream << line;
114              return stream.str();
115          }
116      }
117  }
118
119  string MovieList::TitleString(string title) const
120  {
121      const int MAX = 50; // CALC - Maximum characters per title.
122
123      if (title.length() > MAX)
124      {
125          title = title.substr(0, MAX - 3) + "...";
126      }
127      return title;
128  }
129
130  void MovieList::PrintHeader(ostream& oFile) const
131  {
132      /
          ********************************************************************
          ***
```

```
133     * CONSTANTS
134     *
        ------------------------------------------------------------------------
        -
135     * OUTPUT - USED FOR CLASS HEADING
136     *
        ------------------------------------------------------------------------
        -
137     * PROGRAMMER : Programmer's Name
138     * CLASS      : Student's Course
139     * SECTION    : Class Days and Times
140     * LAB_NUM    : Lab Number (specific to this lab)
141     * LAB_NAME   : Title of the Lab
142     ************************************************************************
        /
143
144     const string AS_NAME = "OPP - DVD Movie List";
145     const int AS_NUM = 5;
146     const string STUDENT_NAME = "Andrew Gharios";
147     const string CLASS_INFO = "M-Th 5-7:20p";
148     const char AS_TYPE = 'A';
149     const long long STUDENT_ID = 1449366;
150
151     oFile << left;
152     oFile << "*************************************************************
        \n";
153     oFile << "*   PROGRAMMED BY : " << STUDENT_NAME << endl;
154     oFile << "*   " << setw(14) << "STUDENT ID " << ": " << STUDENT_ID <<
        endl;
155     oFile << "*   " << setw(14) << "CLASS " << ": " << CLASS_INFO << endl;
156     oFile << "*   ";
157
158     // PROCESSING - This will adjust setws and format appropriately based
159     //              on if this is a lab 'L' or assignment
160
161     if (toupper(AS_TYPE) == 'L')
162     {
163         oFile << "LAB #" << setw(9);
164     }
165     else
166     {
167         oFile << "ASSIGNMENT #" << setw(2);
168     }
169     oFile << AS_NUM << ": " << AS_NAME << endl;
170     oFile << "*************************************************************";
171     oFile << right << endl;
172 }
173
174 /
```

```
      ********************************************************************** ↵
      **
175   * CreateList
176   *   This method will create a list for all the DVDs
177   * INPUTS:
178   *   inFileName : name of input file.
179   *
180   * No outputs.
181   *                                                                      ↵
      ********************************************************************** ↵
      */
182
183   /                                                                      ↵
      ********************************************************************** ↵
      **
184   * DisplayList
185   *   This method will display the entire list of DVDs to the output file.
186   * INPUTS:
187   *   outFileName : output file name.
188   *
189   * No outputs.
190   *                                                                      ↵
      ********************************************************************** ↵
      */
191
192   /                                                                      ↵
      ********************************************************************** ↵
      **
193   * WordWrap
194   *   This method limit the amounts of characters outputted per line.
195   * INPUTS:
196   *   title : Text to word wrap.
197   *
198   * OUPUTS:
199   *   title : Text that is word wrapped.
200   *                                                                      ↵
      ********************************************************************** ↵
      */
201
202   /                                                                      ↵
      ********************************************************************** ↵
      **
203   * TitleString
204   *   This method will make sure the title doesn't exceed the maximum amount.
205   * INPUTS:
206   *   title : Text to word wrap.
207   *
208   * OUPUTS:
209   *   title : Text that is word wrapped.
```

```
210 *                                                                        ↵
    ************************************************************************ ↵
    */
211
212 /                                                                        ↵
    ************************************************************************ ↵
    **
213 * PrintHeader
214 *   This method will output the class header into the outputfile.
215 * INPUTS:
216 *   oFile : Output file ostream variable.
217 *
218 * No outputs.
219 *                                                                        ↵
    ************************************************************************ ↵
    */
220
221
```