

outfile.txt

```
1 *****
2 * PROGRAMMED BY : Faris Hijazi
3 * CLASS       : CS1A
4 * SECTION    : MW: 7:30P
5 * Assignment #5 : DVD Movie List
6 *****
7
8 *****
9 MOVIE # 1   Title: Antwone Fisher
10 -----
11 Year: 2002      Rating:          7
12 -----
13 Leading Actor:  Denzel Washington Genre 1:      Biography
14 Supporting Actor: Derek Luke      Genre 2:      Drama
15 -----
16 PLOT:
17 Antwone Fisher, a young navy man, is forced to see a psychiatrist after a
18 violent outburst against a fellow crewman. During the course of treatment a
19 painful past is revealed and a new hope begins.
20 *****
21
22 *****
23 MOVIE # 2   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 MOVIE # 3   Title: Cinderella Man
37 -----
38 Year: 2005      Rating:          8
39 -----
40 Leading Actor:  Russell Crowe   Genre 1:      Biography
41 Supporting Actor: Renee Zellweger Genre 2:      Drama
42 -----
43 PLOT:
44 The story of James Braddock, a supposedly washed up boxer who came back to
45 become a champion and an inspiration in the 1930s.
46 *****
47
48 *****
49 MOVIE # 4   Title: Five People You Meet in Heaven, The
50 -----
51 Year: 2004      Rating:          7
```

```

52 -----
53 Leading Actor:    Jon Voight      Genre 1:      Drama
54 Supporting Actor: Ellen Burstyn   Genre 2:      Drama
55 -----
56 PLOT:
57 On his 83rd birthday, Eddie (Voight), a war vet and a maintenance worker at
58 the Ruby Pier amusement park, dies while trying to save a girl who is
59 sitting under a falling ride. When he awakens in the afterlife, he
60 encounters five people with ties to his corporeal existence who help him
61 understand the meaning of his life.
62 *****
63
64 *****
65 MOVIE # 5    Title: Good Night, Good Luck
66 -----
67 Year: 2005      Rating:           8
68 -----
69 Leading Actor:   David Strathairn  Genre 1:      Biography
70 Supporting Actor: George Clooney   Genre 2:      Drama
71 -----
72 PLOT:
73 In the early 1950's, the threat of Communism created an air of paranoia in
74 the United States and exploiting those fears was Senator Joseph McCarthy of
75 Wisconsin. However, CBS reporter Edward R. Murrow and his producer Fred W.
76 Friendly decided to take a stand and challenge McCarthy and expose him for
77 the fear monger he was. However, their actions took a great personal toll
78 on both men, but they stood by their convictions and helped to bring down
79 one of the most controversial senators in American history.
80 *****
81
82 *****
83 MOVIE # 6    Title: Invictus
84 -----
85 Year: 2009      Rating:           8
86 -----
87 Leading Actor:   Morgan Freeman    Genre 1:      Biography
88 Supporting Actor: Matt Damon        Genre 2:      Drama
89 -----
90 PLOT:
91 Nelson Mandela, in his first term as the South African President, initiates
92 a unique venture to unite the apartheid-torn land: enlist the national
93 rugby team on a mission to win the 1995 Rugby World Cup.
94 *****
95
96 *****
97 MOVIE # 7    Title: Miss Petigrew Lives for a Day
98 -----
99 Year: 2008      Rating:           7
100 -----
101 Leading Actor:   Frances McDormand Genre 1:      Comedy
102 Supporting Actor: Amy Adams         Genre 2:      Romance

```

103 -----  
 104 PLOT:  
 105 Guinevere Pettigrew, a middle-aged London governess, finds herself unfairly  
 106 dismissed from her job. An attempt to gain new employment catapults her  
 107 into the glamorous world and dizzying social whirl of an American actress  
 108 and singer, Delysia Lafosse.  
 109 \*\*\*\*\*  
 110  
 111 \*\*\*\*\*  
 112 MOVIE # 8 Title: Pay it Forward  
 113 -----  
 114 Year: 2000 Rating: 7  
 115 -----  
 116 Leading Actor: Kevin Spacey Genre 1: Drama  
 117 Supporting Actor: Helen Hunt Genre 2: Romance  
 118 -----  
 119 PLOT:  
 120 Like some other kids, 12-year-old Trevor McKinney believed in the goodness  
 121 of human nature. Like many other kids, he was determined to change the  
 122 world for the better. Unlike most other kids, he succeeded.  
 123 \*\*\*\*\*  
 124  
 125 \*\*\*\*\*  
 126 MOVIE # 9 Title: Pianist, The  
 127 -----  
 128 Year: 2003 Rating: 9  
 129 -----  
 130 Leading Actor: Adrien Brody Genre 1: Biography  
 131 Supporting Actor: Thomas Kreschmann Genre 2: Drama  
 132 -----  
 133 PLOT:  
 134 A Polish Jewish musician struggles to survive the destruction of the Warsaw  
 135 ghetto of World War II.  
 136 \*\*\*\*\*  
 137  
 138 \*\*\*\*\*  
 139 MOVIE # 10 Title: Pirates of the Caribbean 1: The Curse of the Black Pearl  
 140 -----  
 141 Year: 2003 Rating: 8  
 142 -----  
 143 Leading Actor: Johnny Depp Genre 1: Action  
 144 Supporting Actor: Orlando Bloom Genre 2: Action  
 145 -----  
 146 PLOT:  
 147 This swash-buckling tale follows the quest of Captain Jack Sparrow, a savvy  
 148 pirate, and Will Turner, a resourceful blacksmith, as they search for  
 149 Elizabeth Swann. Elizabeth, the daughter of the governor and the love of  
 150 Will's life, has been kidnapped by the feared Captain Barbossa. Little do  
 151 they know, but the fierce and clever Barbossa has been cursed. He, along  
 152 with his large crew, is under an ancient curse, doomed for eternity to  
 153 neither live, nor die. That is, unless a blood sacrifice is made. Full of

154 edge-of-the-seat action and swashbuckling adventures, this is a movie you  
 155 won't want to miss!

156 \*\*\*\*\*  
 157

158 \*\*\*\*\*  
 159 MOVIE # 11 Title: Pursuit of Happyness  
 160 -----  
 161 Year: 2008 Rating: 8  
 162 -----  
 163 Leading Actor: Will Smith Genre 1: Biography  
 164 Supporting Actor: Jaden Smith Genre 2: Drama  
 165 -----  
 166 PLOT:  
 167 A struggling salesman takes custody of his son as he's poised to begin a  
 168 life-changing professional endeavor.♦  
 169 \*\*\*\*\*  
 170

171 \*\*\*\*\*  
 172 MOVIE # 12 Title: Rent  
 173 -----  
 174 Year: 2005 Rating: 7  
 175 -----  
 176 Leading Actor: Anthony Rapp Genre 1: Musical  
 177 Supporting Actor: Adam Pascal Genre 2: Drama  
 178 -----  
 179 PLOT:  
 180 This is the film version of the Pulitzer and Tony Award winning musical  
 181 about Bohemians in the East Village of New York City struggling with life,  
 182 love and AIDS, and the impacts they have on America.  
 183 \*\*\*\*\*  
 184

185 \*\*\*\*\*  
 186 MOVIE # 13 Title: Secondhand Lions  
 187 -----  
 188 Year: 2003 Rating: 8  
 189 -----  
 190 Leading Actor: Haley Joel Osmet Genre 1: Comedy  
 191 Supporting Actor: Robert Duvall Genre 2: Comedy  
 192 -----  
 193 PLOT:  
 194 A coming-of-age story about a shy, young boy sent by his irresponsible  
 195 mother to spend the summer with his wealthy, eccentric uncles in Texas.  
 196 \*\*\*\*\*  
 197

198 \*\*\*\*\*  
 199 MOVIE # 14 Title: Shackelton  
 200 -----  
 201 Year: 2002 Rating: 8  
 202 -----  
 203 Leading Actor: Kenneth Branagh Genre 1: Biography  
 204 Supporting Actor: John Grillo Genre 2: Drama

205 -----  
 206 PLOT:  
 207 The true story of Shackleton's 1914 Endurance expedition to the the South  
 208 Pole and his epic struggle to lead his 28 man crew to safety after his ship  
 209 was crushed in the pack ice.  
 210 \*\*\*\*\*  
 211  
 212 \*\*\*\*\*  
 213 MOVIE # 15 Title: Shawshank Redemption, The  
 214 -----  
 215 Year: 1994 Rating: 9  
 216 -----  
 217 Leading Actor: Tim Robbins Genre 1: Drama  
 218 Supporting Actor: Morgan Freeman Genre 2: Crime  
 219 -----  
 220 PLOT:  
 221 Two imprisoned men bond over a number of years, finding solace and eventual  
 222 redemption through acts of common decency.  
 223 \*\*\*\*\*  
 224  
 225 \*\*\*\*\*  
 226 MOVIE # 16 Title: Thelma and Louise  
 227 -----  
 228 Year: 1991 Rating: 7  
 229 -----  
 230 Leading Actor: Susan Sarandon Genre 1: Action  
 231 Supporting Actor: Geena Davis Genre 2: Action  
 232 -----  
 233 PLOT:  
 234 Whilst on a short weekend getaway, Louise shoots a man who had tried to  
 235 rape Thelma. Due to the incriminating circumstances, they make a run for it  
 236 but are soon followed closely by the authorities including a local  
 237 policeman who is sympathetic to their plight. The federal authorities,  
 238 however, have less compassion and thus a cross country chase ensues for the  
 239 two fugitives. Along the way, both women rediscover the strength of their  
 240 friendship and surprising aspects of their personalities and self-strengths  
 241 in the trying times.  
 242 \*\*\*\*\*  
 243  
 244 \*\*\*\*\*  
 245 MOVIE # 17 Title: Unfinished Life, An  
 246 -----  
 247 Year: 2005 Rating: 7  
 248 -----  
 249 Leading Actor: Robert Redford Genre 1: Drama  
 250 Supporting Actor: Jennifer Lopez Genre 2: Drama  
 251 -----  
 252 PLOT:  
 253 A down on her luck woman, desperate to provide care for her daughter, moves  
 254 in with her father in-law from whom she is estranged. Through time, they  
 255 learn to forgive each other and heal old wounds.

```

256 *****
257
258 *****
259 MOVIE # 18 Title: Whip It
260 -----
261 Year: 2009 Rating: 7
262 -----
263 Leading Actor: Ellen Page Genre 1: Drama
264 Supporting Actor: Marcia Gay Harden Genre 2: Sport
265 -----
266 PLOT:
267 In Bodeen, Texas, an indie-rock loving misfit finds a way of dealing with
268 her small-town misery after she discovers a roller derby league in nearby Austin.
269 *****
270
271 *****
272 MOVIE # 19 Title: X-Men
273 -----
274 Year: 2000 Rating: 7
275 -----
276 Leading Actor: Hugh Jackman Genre 1: Action
277 Supporting Actor: Patrick Stewart Genre 2: Action
278 -----
279 PLOT:
280 All over the planet, unusual children are born with an added twist to their
281 genetic code. This X-factor allows the children to perform extraordinary
282 feats - flight, telekinetics, laser beams from the eyes and more. One Dr.
283 Charles Xavier gathers the children to a place where he can train them to
284 use their powers for themselves and the forces of good he dubs these
285 children his X-men and hijinks ensue.
286 *****
287
288 *****
289 MOVIE # 20 Title: Yankee Doodle Dandee
290 -----
291 Year: 1942 Rating: 8
292 -----
293 Leading Actor: James Cagney Genre 1: Musical
294 Supporting Actor: Joan Leslie Genre 2: Biography
295 -----
296 PLOT:
297 This film depicts the life of the renowned musical composer, playwright,
298 actor, dancer and singer George M. Cohan.
299 *****
300
301

```

output.txt

1 which output file would you like to use (d for default) ?: d  
2 which input file would you like to use (d for default) ?: d

mainheader.h

```
1  /*****
2  * AUTHOR      : Faris Hijazi
3  * STUDENT ID   : 1039438
4  * ASSIGNMENT 5 : DVD Movie List Intro to OOP
5  * CLASS       : CS1A
6  * SECTION     : MW 7:30PM
7  * DUE DATE    : 05/21/19
8  *****/
9
10 #ifndef HEADER_H_
11 #define HEADER_H_
12 #include <string>
13 #include <iostream>
14 #include <iomanip>
15 #include <string>
16 #include <fstream>
17 using namespace std;
18
19 struct DVDNode
20 {
21     string title;           // store movie's title
22     string lActor;          // store leading actor/actress
23     string sActor;          // store supporting actor/actress
24     string genre;           // store movie's genre
25     string sGenre;          // store movie's alternate genre
26     int year;               // store movie's release year
27     int rating;             // store movie's rating
28     string synopsis;        // store movie's synopsis
29     DVDNode *next;          // linked list next pointer
30 };
31
32 #endif /* HEADER_H_ */
```



main.cpp

```
1  /*****
2  * AUTHOR      : Faris Hijazi
3  * STUDENT ID   : 1039438
4  * ASSIGNMENT 5 : DVD Movie List Intro to OOP
5  * CLASS       : CS1A
6  * SECTION     : MW 7:30PM
7  * DUE DATE    : 05/21/19
8  *****/
9
10 #include "classheader.h"
11
12 int main()
13 {
14     MovieList movieList;    //IN&OUT - instance of MovieList class
15     string input;           //IN      - file name input
16     string offileName;      //IN      - name of output file
17     string iffileName;      //IN      - name of input file
18
19     //get names of in and out files
20     cout << "which output file would you like to use (d for default) ? : ";
21     getline(cin, input); //name output file
22     if(input == "d")
23     {
24         offileName = "outfile.txt";
25     }
26     else
27     {
28         offileName = input;
29     }
30
31     cout << "which input file would you like to use (d for default) ? : ";
32     getline(cin, iffileName); //name output file
33     if(input == "d")
34     {
35         iffileName = "inputfile.txt";
36     }
37     else
38     {
39         iffileName = input;
40     }
41     cout << endl;
42
43     //create and output list
44     movieList.CreateList(iffileName);
45     movieList.OutputList(offileName);
46
47     return 0;
48 }
```

classheader.h

```
1 #ifndef CLASS_HEADER_H_
2 #define CLASS_HEADER_H_
3 #include "mainheader.h"
4 /*****
5  * AUTHOR      : Faris Hijazi
6  * STUDENT ID   : 1039438
7  * ASSIGNMENT 5 : DVD Movie List Intro to OOP
8  * CLASS       : CS1A
9  * SECTION     : MW 7:30PM
10 * DUE DATE    : 05/21/19
11 *****/
12
13 class StackList // Base Class - Using linked list implementation
14 {
15     public:
16         /**
17          * CONSTRUCTOR
18          */
19         StackList();
20         /**
21          * DECONSTRUCTOR
22          */
23         ~StackList();
24
25         /**
26          * MUTATORS
27          */
28         void Push(DVDNode newDVD); // create a DVDNode, add a
29                                     // DVDNode in the stack, by adding
30                                     // to the front of the linked List
31
32         DVDNode Pop(); // return the DVDNode in the top of
33                       // the stack, remove the DVDNode
34                       // from the stack, delete the DVDNode
35         /**
36          * ACCESSORS
37          */
38         bool IsEmpty() const; // check if stack is empty
39         DVDNode Peek() const; // return the DVDNode at the top of the stack
40         int Size() const; // return the number of people in the stack
41
42     protected:
43         DVDNode *head; // head pointer for stack
44         int stackCount; // total number of persons in the stack
45 };
46
47 class MovieList: public StackList // Derived Class
48 {
49     public:
50         /**
51          * CONSTRUCTOR
```

```

52     **/
53     MovieList();
54     /*
55     * DESTRUCTOR
56     **/
57     ~MovieList();
58
59     /**
60     * ACCESSORS
61     **/
62     void OutputList(string outFileName) const;
63     /**
64     * MUTATORS
65     **/
66     void CreateList(string inFileName);
67 private:
68     void wordwrap(ostream &output, DVDNode *&DVDPtr) const;
69     void PrintHeader(ostream &output, //output device to use
70                     char exersize, //assignment or lab
71                     string exersizeName, //name of exersize
72                     int num, //as or lab number
73                     string names) const; //names;
74 };
75
76 #endif

```

methods.cpp

```
1  /*****
2  * AUTHOR      : Faris Hijazi
3  * STUDENT ID   : 1039438
4  * ASSIGNMENT 5 : DVD Movie List Intro to OOP
5  * CLASS       : CS1A
6  * SECTION     : MW 7:30PM
7  * DUE DATE    : 05/21/19
8  *****/
9
10 #include "classheader.h"
11
12 /**
13  * StackList Methods
14  */
15
16 StackList::StackList()
17 {
18     stackCount = 0;
19     head = NULL;
20 }
21
22 StackList::~StackList()
23 {
24     DVDNode *DVDPtr;    //pointer to keep track of nodes
25
26     while(head != NULL)
27     {
28         DVDPtr = head;
29         head = DVDPtr->next;
30         delete DVDPtr;
31     }
32 }
33
34 /*****
35  * This method will add a node to a linked list
36  *-----
37  * INPUT:
38  *     newDVD - new node to add
39  * OUTPUT:
40  *     NA
41  *****/
42 void StackList::Push(DVDNode newDVD)
43 {
44     DVDNode *DVDPtr;    //pointer to keep track of nodes
45
46     DVDPtr = new DVDNode;
47     if(DVDPtr != NULL)
48     {
49         *DVDPtr = newDVD;
50         DVDPtr->next = head;
51         head = DVDPtr;
```

```

52     DVDPtr = NULL;
53 }
54     stackCount++;
55 }
56
57 /**
58  * MovieList Methods
59  */
60 MovieList::MovieList()
61 {}
62 MovieList::~MovieList()
63 {}
64
65 /*****
66  * This method will create a linked list
67  *-----
68  * INPUT:
69  *     inFileName - name of input file
70  * OUTPUT:
71  *     NA
72  *****/
73 void MovieList::CreateList(string inFileName)
74 {
75     ifstream infile;    //input file name
76     DVDNode newNode;    //pointer to keep track of nodes
77
78     infile.open(inFileName);
79     while(infile)
80     {
81         getline(infile, newNode.title);
82         getline(infile, newNode.lActor);
83         getline(infile, newNode.sActor);
84         getline(infile, newNode.genre);
85         getline(infile, newNode.sGenre);
86         infile >> newNode.year;
87         infile.ignore(1000, '\n');
88         infile >> newNode.rating;
89         infile.ignore(1000, '\n');
90         getline(infile, newNode.synopsis);
91         infile.ignore(1000, '\n');
92         StackList::Push(newNode);
93     }
94     infile.close();
95 }
96
97 /*****
98  * This method will output a linked list to a specified output file
99  *-----
100 * INPUT:
101 *     outFileName - name of output file
102 * OUTPUT:

```

```

103 *      formatted linked list
104 *****/
105
106 void MovieList::OutputList(string outFileName) const
107 {
108     const int COL_SIZE_TITLE = 18; //col width
109     const int MAX_WIDTH_TITLE = 75; //col width
110
111     int i = 0;          //counter for nodes outputted
112     ofstream ofile;     //output file variable
113     DVDNode *DVDPtr;    //pointer to keep track of nodes
114
115     ofile.open(outFileName);
116
117     PrintHeader(ofile, 'A', "DVD Movie List", 5, "Faris Hijazi");
118     DVDPtr = head;
119     while(DVDPtr != NULL)
120     {
121         ofile << left;
122         ofile << setfill('*') << setw(MAX_WIDTH_TITLE)
123             << '*' << endl;
124         ofile << setfill(' ');
125         ofile << setw(COL_SIZE_TITLE-6) << "MOVIE # " + to_string(++i);
126         ofile << "Title: " << DVDPtr->title << endl;
127         ofile << setfill('-') << setw(MAX_WIDTH_TITLE) << '-' << endl;
128         ofile << setfill(' ') << "Year: " << setw(COL_SIZE_TITLE-6)
129             << DVDPtr->year;
130         ofile << setw(COL_SIZE_TITLE) << "Rating: "
131             << DVDPtr->rating << endl;
132         ofile << setfill('-') << setw(MAX_WIDTH_TITLE)
133             << '-' << endl;
134         ofile << setfill(' ') << setw(COL_SIZE_TITLE)
135             << "Leading Actor:" << setw(COL_SIZE_TITLE)
136             << DVDPtr->lActor << setw(COL_SIZE_TITLE)
137             << "Genre 1: " << DVDPtr->genre << endl;
138         ofile << setw(COL_SIZE_TITLE)
139             << "Supporting Actor:" << setw(COL_SIZE_TITLE)
140             << DVDPtr->sActor << setw(COL_SIZE_TITLE)
141             << "Genre 2: " << DVDPtr->sGenre << endl;
142         ofile << setfill('-') << setw(MAX_WIDTH_TITLE) << '-';
143         ofile << "\nPLOT:\n";
144         wordwrap(ofile, DVDPtr);
145         ofile << setfill('*') << setw(MAX_WIDTH_TITLE)
146             << '*' << endl << endl;
147         ofile << setfill(' ');
148         DVDPtr = DVDPtr->next;
149     }
150     ofile.close();
151 }
152
153

```

```

154 /*****
155 * This method will word wrap a given line to 75 characters per line
156 *-----
157 * INPUT:
158 *     output - output device
159 *     DVDPtr - Pointer to struct to access string member
160 * OUTPUT:
161 *     word wrapped string
162 *****/
163 void MovieList::wordwrap(ostream &output, DVDNode *&DVDPtr) const
164 {
165     const int MAX = 75; //max width of line
166     int i;              //for loop lcv
167     int j;              //for loop lcv
168     string word;         //individual word
169     string line;         //individual line
170     for(i=0; i < DVDPtr->synopsis.size(); i++)
171     {
172         if(DVDPtr->synopsis[i] != ' ')
173         {
174             word += DVDPtr->synopsis[i];
175         }
176         else
177         {
178             if (word.size() + line.size() > MAX)
179             {
180                 output << line;
181                 output << endl;
182                 line.clear();
183             }
184             line += word + ' ';
185             word.clear();
186         }
187         if(i == DVDPtr->synopsis.size()-1)
188         {
189             output << line << word << endl;
190         }
191     }
192 }
193
194 /*****
195 * This function will output the class header using ostream
196 *-----
197 * INPUT:
198 *     output - output file variable
199 *     exersize - Lab or Assignment
200 *     exersizeName- name of exersize
201 *     num - number of Lab/Assignment
202 *     names - names of programmers
203 * OUTPUT:
204 *     header

```

```

205 *****/
206 void MovieList::PrintHeader(ostream &output, char exersize, string exersizeName, int
    num, string names) const
207 {
208
209     int colWidth; //CALC - changes based on exersize
210     string asType; //CALC - changes based on exersize
211
212     if(exersize == 'L')
213     {
214         asType = "Lab";
215         colWidth = 9;
216     }
217     else
218     {
219         asType = "Assignment";
220         colWidth = 2;
221     }
222
223     output << left;
224     output << "*****\n";
225     output << "* PROGRAMMED BY : " << names << endl;
226     output << "* " << setw(14) << "CLASS" << ": " << "CS1A" << endl;
227     output << "* " << setw(14) << "SECTION" << ": " << "MW: 7:30P" << endl;
228     output << "* " << asType << " #" << setw(colWidth) << num << ": " << exersizeName
    << endl;
229     output << "*****\n\n";
230     output << right;
231 }
232

```