

iostream.txt

```
1 *****
2 * PROGRAMMED BY : ALI ESHGHI
3 * CLASS : CS1B
4 * SECTION : MW: 7:30p - 9:50p
5 * LAB #13 : LAB 13 - ARRAYS AND LINKED LIST (OOP)
6 *****
7
8 *****
9 * WELCOME TO THE SHEEP LIST MANAGER *
10 *****
11
12 SHEEP LIST MANAGER
13 1 - Add Sheep
14 2 - Output 1st Sheep
15 3 - Find Sheep
16 4 - List Size
17 5 - Output List
18 6 - Clear list
19 0 - Exit
20 Enter a command: 1
21 Sheep name: Fluffy
22 Sheep Age: 1
23
24
25 The Sheep...
26 Sheep Name: Fluffy
27 Sheep Age: 1
28 Has been added
29
30 SHEEP LIST MANAGER
31 1 - Add Sheep
32 2 - Output 1st Sheep
33 3 - Find Sheep
34 4 - List Size
35 5 - Output List
36 6 - Clear list
37 0 - Exit
38 Enter a command: 2
39 NAME AGE
40 -----
41 Fluffy 1
42
43
44 Is at the front of the list!
45
46 SHEEP LIST MANAGER
47 1 - Add Sheep
48 2 - Output 1st Sheep
49 3 - Find Sheep
50 4 - List Size
51 5 - Output List
52 6 - Clear list
53 0 - Exit
54 Enter a command: 1
55 Sheep name: Maa
```

```

56 Sheep Age: 3
57
58
59 The Sheep...
60 Sheep Name: Maa
61 Sheep Age: 3
62 Has been added
63
64 SHEEP LIST MANAGER
65 1 - Add Sheep
66 2 - Output 1st Sheep
67 3 - Find Sheep
68 4 - List Size
69 5 - Output List
70 6 - Clear list
71 0 - Exit
72 Enter a command: 4
73 There are 2 sheeps in the list
74
75 SHEEP LIST MANAGER
76 1 - Add Sheep
77 2 - Output 1st Sheep
78 3 - Find Sheep
79 4 - List Size
80 5 - Output List
81 6 - Clear list
82 0 - Exit
83 Enter a command: 5
84 <output using the array>
85 NAME          AGE
86 -----
87 Fluffy         1
88 Maa            3
89
90
91 <output using the linked list>
92 NAME          AGE
93 -----
94 Fluffy         1
95 Maa            3
96 There are 2 sheeps in the list
97
98 SHEEP LIST MANAGER
99 1 - Add Sheep
100 2 - Output 1st Sheep
101 3 - Find Sheep
102 4 - List Size
103 5 - Output List
104 6 - Clear list
105 0 - Exit
106 Enter a command: 1
107 Sheep name: Baa Baa
108 Sheep Age: 2
109
110

```

```

111 The Sheep...
112 Sheep Name: Baa Baa
113 Sheep Age: 2
114 Has been added
115
116 SHEEP LIST MANAGER
117 1 - Add Sheep
118 2 - Output 1st Sheep
119 3 - Find Sheep
120 4 - List Size
121 5 - Output List
122 6 - Clear list
123 0 - Exit
124 Enter a command: 5
125 <output using the array>
126 NAME          AGE
127 -----
128 Fluffy         1
129 Maa            3
130 Baa Baa       2
131
132
133 <output using the linked list>
134 NAME          AGE
135 -----
136 Fluffy         1
137 Maa            3
138 Baa Baa       2
139 There are 3 sheeps in the list
140
141 SHEEP LIST MANAGER
142 1 - Add Sheep
143 2 - Output 1st Sheep
144 3 - Find Sheep
145 4 - List Size
146 5 - Output List
147 6 - Clear list
148 0 - Exit
149 Enter a command: 4
150 There are 3 sheeps in the list
151
152 SHEEP LIST MANAGER
153 1 - Add Sheep
154 2 - Output 1st Sheep
155 3 - Find Sheep
156 4 - List Size
157 5 - Output List
158 6 - Clear list
159 0 - Exit
160 Enter a command: 3
161 Who are you looking for? Baa Baa
162
163 NAME          AGE
164 -----
165

```

```
166 Baa Baa          2
167
168 Has Been Found!
169
170 SHEEP LIST MANAGER
171 1 - Add Sheep
172 2 - Output 1st Sheep
173 3 - Find Sheep
174 4 - List Size
175 5 - Output List
176 6 - Clear list
177 0 - Exit
178 Enter a command: 6
179
180 The list has been cleared!
181
182 SHEEP LIST MANAGER
183 1 - Add Sheep
184 2 - Output 1st Sheep
185 3 - Find Sheep
186 4 - List Size
187 5 - Output List
188 6 - Clear list
189 0 - Exit
190 Enter a command: 6
191
192 The list is empty
193
194 SHEEP LIST MANAGER
195 1 - Add Sheep
196 2 - Output 1st Sheep
197 3 - Find Sheep
198 4 - List Size
199 5 - Output List
200 6 - Clear list
201 0 - Exit
202 Enter a command: 5
203
204 The list is empty
205
206
207 SHEEP LIST MANAGER
208 1 - Add Sheep
209 2 - Output 1st Sheep
210 3 - Find Sheep
211 4 - List Size
212 5 - Output List
213 6 - Clear list
214 0 - Exit
215 Enter a command: 4
216
217 The list is empty
218
219
220 SHEEP LIST MANAGER
```

```
221 1 - Add Sheep
222 2 - Output 1st Sheep
223 3 - Find Sheep
224 4 - List Size
225 5 - Output List
226 6 - Clear list
227 0 - Exit
228 Enter a command: 3
229
230 There are no sheep to be found!
231
232 SHEEP LIST MANAGER
233 1 - Add Sheep
234 2 - Output 1st Sheep
235 3 - Find Sheep
236 4 - List Size
237 5 - Output List
238 6 - Clear list
239 0 - Exit
240 Enter a command: 2
241 Nobody is in front -the list is empty!
242
243 SHEEP LIST MANAGER
244 1 - Add Sheep
245 2 - Output 1st Sheep
246 3 - Find Sheep
247 4 - List Size
248 5 - Output List
249 6 - Clear list
250 0 - Exit
251 Enter a command: 7
252
253 **** The number 7 is an invalid entry      ****
254 **** Please input a number between 0 or 6 ****
255
256 SHEEP LIST MANAGER
257 1 - Add Sheep
258 2 - Output 1st Sheep
259 3 - Find Sheep
260 4 - List Size
261 5 - Output List
262 6 - Clear list
263 0 - Exit
264 Enter a command: 0
265
266
```

MyHeader.h

```
1 /*****
2 * AUTHOR      : Ali Eshghi
3 * STUDENT ID   : 1112261
4 * LAB #13      : LAB 13 - ARRAYS AND LINKED LIST (OOP)
5 * CLASS       : CS 1B
6 * SECTION     : MW - 7:30 pm - 9:50 pm
7 * DUE DATE    : 12/3/2019
8 *****/
9
10 #ifndef MYHEADER_H_
11 #define MYHEADER_H_
12
13 #include<iostream>
14 #include<iomanip>
15 #include<string>
16 #include<fstream>
17 #include<limits>
18 #include<sstream>
19 #include "ClassHeader.h"
20 using namespace std;
21
22 /*****
23 * CONSTANTS
24 * -----
25 * USED FOR CLASS HEADING - ALL WILL BE OUTPUT
26 * -----
27 * Type      : Program Type
28 * LAB_NUM   : Lab Number (specific to this lab)
29 * LAB_NAME  : Title of the Lab
30 *****/
31
32 const string NAME = "LAB 13 - ARRAYS AND LINKED LIST (OOP)";
33 const char   TYPE  = 'L';
34 const int    NUM   = 13 ;
35 const string CLASS = "CS1B";
36 const string SECTION = "MW: 7:30p - 9:50p";
37
38 /*****
39 * Function - PrintHeaderFile
40 * -----
41 * This function will output the class heading to the screen.
42 *
43 * return type - nothing
44 *              the function is void type
45 *****/
46 void PrintHeader();
47
48 /*****
49 * Function - Menu
50 * -----
51 * This function will outputs the menu and prompts the user
52 * to choose an option from the menu
53 *
54 * return type - integer
55 *              the function is int type
```

MyHeader.h

```
56  *****/
57  int Menu();
58
59  #endif /* MYHEADER_H_ */
60
```

main.cpp

```

1 /*****
2 * AUTHOR      : Ali Eshghi
3 * STUDENT ID   : 1112261
4 * LAB #13      : LAB 13 - ARRAYS AND LINKED LIST (OOP)
5 * CLASS        : CS 1B
6 * SECTION      : MW - 7:30 pm - 9:50 pm
7 * DUE DATE     : 12/3/2019
8 *****/
9 #include "MyHeader.h"
10 #include "Classheader.h"
11
12 /*****
13 * LAB 13 - ARRAYS AND LINKED LIST (OOP)
14 *
15 * This program will use the arrays and linked list as an
16 * objects of a class and gets the information base of the
17 * users choice and add those information to the list and
18 * prompts a menu for the user who can have more options
19 *
20 * INPUT:menuOption -> user input for the menu
21 *      name       -> name of a new sheep
22 *      age        -> age of a new sheep
23 *
24 *
25 * OUTPUT:name      -> name of the sheeps in the list
26 *      age         -> age of the sheeps in the list
27 *      size        -> how many sheeps are in the list
28 *      search-> user input name to search in the list
29 *      if the list is empty or no
30 *****/
31
32 int main()
33 {
34
35     /*****
36     * VARIABLE *
37     *****/
38     int menuOption;
39     Animal sheep;
40     string name;
41     string search;
42     int age;
43     int size;
44
45
46     //this function will output the class header
47     PrintHeader();
48
49
50
51     cout << "*****" << endl;
52     cout << "* WELCOME TO THE SHEEP LIST MANAGER *" << endl;
53     cout << "*****" << endl << endl;
54
55     //do while loop for menu

```



```

56  do
57  {
58      //this function prompts the user the menu and gets the user input
59      menuOption = Menu();
60
61
62      //first option that adds the sheep to the menu
63      if(menuOption == 1)
64      {
65          cin.ignore(10000, '\n');
66          cout << "Sheep name: ";
67          getline(cin, name);
68          cout << "Sheep Age: ";
69          cin >> age;
70
71          sheep.AddSheep(name, age);
72          sheep.AddSheepLinkedList(name, age);
73      }
74
75      //second option that displays the first sheep in the list
76      else if(menuOption == 2)
77      {
78          sheep.DisplayFirstSheep();
79      }
80
81      //third option that searches for a specific sheep
82      else if(menuOption == 3)
83      {
84          cin.ignore(10000, '\n');
85          cout << "Who are you looking for? ";
86          getline(cin, search);
87          sheep.FindSheep(search);
88      }
89
90      //fourth option that outputs how many sheeps are in the list
91      else if(menuOption == 4)
92      {
93          size = sheep.ListSize();
94          cout << "There are " << size << " sheeps in the list"
95               << endl << endl;
96      }
97
98      //fifth option that displays the members of the list
99      else if(menuOption == 5)
100     {
101         sheep.Display();
102
103         size = sheep.ListSize();
104         cout << "There are " << size << " sheeps in the list"
105              << endl << endl;
106     }
107
108     //sixth option that clears the list (deconstruction)
109     else if(menuOption == 6)
110     {

```

main.cpp

```
111         sheep.~Animal();
112
113     }
114
115
116     }while(menuOption != 0);
117
118
119     return 0;
120
121
122 }
123
```

Menu.cpp

```
1 /*****
2 * AUTHOR      : Ali Eshghi
3 * STUDENT ID   : 1112261
4 * LAB #13      : LAB 13 - ARRAYS AND LINKED LIST (OOP)
5 * CLASS        : CS 1B
6 * SECTION      : MW - 7:30 pm - 9:50 pm
7 * DUE DATE     : 12/3/2019
8 *****/
9
10 #include "MyHeader.h"
11
12 /*****
13 * Function - Menu
14 * -----
15 * This function will outputs the menu and prompts the user
16 * to choose an option from the menu
17 *
18 * return type - integer
19 *               the function is int type
20 *****/
21
22 int Menu()
23 {
24
25     /*****
26     * VARIABLE *
27     *****/
28
29     int menuOption;
30     bool checkInp;
31
32     /*****
33     * INITIALIZATION *
34     *****/
35
36     checkInp = false;
37
38     //do while loop for input check
39     do
40     {
41
42         cout << "SHEEP LIST MANAGER" << endl;
43         cout << "1 - Add Sheep" << endl;
44         cout << "2 - Output 1st Sheep" << endl;
45         cout << "3 - Find Sheep" << endl;
46         cout << "4 - List Size" << endl;
47         cout << "5 - Output List" << endl;
48         cout << "6 - Clear list" << endl;
49         cout << "0 - Exit" << endl;
50         cout << "Enter a command: ";
51
52
53
54         //CHECKS FOR THE CHAR INPUT
55
```

Menu.cpp

```
56  if (!(cin >> menuOption))
57  {
58      cin.clear();
59      cin.ignore(numeric_limits<streamsize>::max(), '\n');
60
61      cout << endl;
62      cout << "**** Please input a NUMBER between 0 or 6 ****";
63      cout << endl << endl;
64
65      checkInp = false;
66
67  }
68
69  //CHECKS FOR THE RANGE ERROR
70
71  else if (menuOption > 6 || menuOption < 0 )
72  {
73
74      cout << endl;
75      cout << "**** The number " << menuOption
76      << " is an invalid entry ****" << endl;
77      cout << "**** Please input a number between 0 or 6 ****";
78      cout << endl << endl;
79
80      checkInp = false;
81
82  }
83
84  checkInp = true;
85
86  }while(!checkInp);
87
88
89  return menuOption;
90 }
91
92
93
```

PrintHeader.cpp

```
1 /*****
2 * AUTHOR      : Ali Eshghi
3 * STUDENT ID   : 1112261
4 * LAB #13      : LAB 13 - ARRAYS AND LINKED LIST (OOP)
5 * CLASS       : CS 1B
6 * SECTION     : MW - 7:30 pm - 9:50 pm
7 * DUE DATE    : 12/3/2019
8 *****/
9
10 #include "MyHeader.h"
11
12 /*****
13 * Function - PrintHeaderFile
14 * -----
15 * This function will output the class heading to the screen.
16 *
17 * return type - nothing
18 *             the function is void type
19 *****/
20
21 void PrintHeader()
22 {
23     cout << left;
24     cout << "*****\n" ;
25     cout << "* PROGRAMMED BY : ALI ESHGHI" ;
26     cout << "\n* " << setw(14) << "CLASS" << ": " << CLASS ;
27     cout << "\n* " << setw(14) << "SECTION" << ": " << SECTION ;
28     cout << "\n* LAB #" << setw(9) << NUM << ": " << NAME ;
29     cout << "\n*****\n\n" ;
30     cout << right;
31 }
32
33
```

ClassHeader.h

```
1 /*****
2 * AUTHOR      : Ali Eshghi
3 * STUDENT ID   : 1112261
4 * LAB #13      : LAB 13 - ARRAYS AND LINKED LIST (OOP)
5 * CLASS        : CS 1B
6 * SECTION      : MW - 7:30 pm - 9:50 pm
7 * DUE DATE     : 12/3/2019
8 *****/
9
10 #ifndef CLASSHEADER_H_
11 #define CLASSHEADER_H_
12
13 #include<iostream>
14 #include<iomanip>
15 #include<string>
16 #include<fstream>
17 #include<limits>
18 #include<sstream>
19 using namespace std;
20
21 const int AR_SIZE = 50;
22
23
24 class Animal
25 {
26     //public part of the class that is available for outside of the class
27     public:
28         //constructor
29         Animal();
30
31         //destructor
32         ~Animal();
33
34         //method for adding a new sheep and its age to parallel arrays
35         void AddSheep(string name, int age);
36
37         //method for adding a new sheep to a linked list
38         void AddSheepLinkedList(string name, int age);
39
40         //method for showing the first sheep from the list
41         void DisplayFirstSheep();
42
43         //method that returns the size of the list of the sheeps
44         int ListSize() const;
45
46         //method for outputting the objects
47         void Display() const;
48
49         //method for finding the sheep in the list
50         void FindSheep(string) const;
51
52
53     //private part only available for the class
54     private:
55         string nameAr[AR_SIZE];
```

ClassHeader.h

```
56     int    ageAr[AR_SIZE];
57     string name;
58     int    age;
59     int    listSize;
60
61     struct SheepNode
62     {
63         string    sheepName;
64         int       sheepAge;
65         SheepNode *next;
66     };
67     SheepNode *head;
68
69 };
70
71
72
73 #endif /* CLASSHEADER_H_ */
74
```

Methods.cpp

```
1 /*****
2 * AUTHOR      : Ali Eshghi
3 * STUDENT ID   : 1112261
4 * LAB #13      : LAB 13 - ARRAYS AND LINKED LIST (OOP)
5 * CLASS        : CS 1B
6 * SECTION      : MW - 7:30 pm - 9:50 pm
7 * DUE DATE     : 12/3/2019
8 *****/
9
10 #include "MyHeader.h"
11 #include "ClassHeader.h"
12
13
14 Animal::Animal()/**/ CONSTRUCTOR ***/
15 {
16     /*****
17      *  INITIALIZATION *
18      *****/
19
20     name.clear();
21     age = 0;
22     listSize = 0;
23     head = NULL;
24 }
25
26 Animal::~Animal()    /**/ DESTRUCTOR ***/
27 {
28
29     /*****
30      *  VARIABLE *
31      *****/
32
33     SheepNode *sheepPtr;
34
35     if(head != NULL)
36     {
37         //clear the list
38         sheepPtr = head;
39         while(sheepPtr != NULL)
40         {
41             head = head -> next;
42             delete sheepPtr;
43
44             sheepPtr = head;
45         }
46
47         for(int i = 0; i < AR_SIZE; i++)
48         {
49             nameAr[i] = ' ';
50         }
51         cout << "The list has been cleared!" << endl << endl;
52     }
53
54     else if(head == NULL)
55     {
```



```

56     cout << "\nThe list is empty" << endl << endl;
57 }
58
59 }
60
61
62 //method for adding a new sheep and its age to parallel arrays
63 void Animal::AddSheep(string name, int age)
64 {
65     if(listSize < AR_SIZE)
66     {
67         nameAr[listSize] = name;
68         ageAr[listSize] = age;
69
70         listSize++;
71
72         cout << endl << endl;
73         cout << "The Sheep..." << endl;
74         cout << "Sheep Name: " << name << endl;
75         cout << "Sheep Age: " << age << endl;
76         cout << "Has been added" << endl << endl;
77     }
78
79     else
80     {
81         cout << "could not add new animal, list is full..." << endl;
82     }
83
84 }
85
86 //method for adding a new sheep to a linked list
87 void Animal::AddSheepLinkedList(string name, int age)
88 {
89     /*****
90      *   VARIABLE *
91      *****/
92
93     SheepNode *newSheepNode;
94     SheepNode *tail;
95
96     /*****
97      *   INITIALIZATION *
98      *****/
99
100    newSheepNode = new SheepNode;
101
102    /*** ADD TO THE TAIL ***
103
104    //check if there is memory for new node
105    if(newSheepNode != NULL)
106    {
107        newSheepNode -> sheepName = name;
108        newSheepNode -> sheepAge = age;
109
110

```

```

111     //check if list is empty
112     if(head != NULL)
113     {
114         tail = head;
115
116         //find the tail
117         while(tail != NULL)
118         {
119             tail = tail -> next;
120         }
121
122         tail -> next = newSheepNode;
123     }
124
125     else
126     {
127         head = newSheepNode;
128     }
129
130     listSize++;
131 }
132
133 else
134 {
135     cout << "Could not add to the list - out of memory";
136 }
137 }
138
139 //method that returns the size of the list of the sheeps
140 int Animal::ListSize() const
141 {
142     if(head != NULL)
143     {
144         return listSize;
145     }
146     else if(head == NULL)
147     {
148         cout << "\nThe list is empty" << endl << endl;
149         return 0;
150     }
151 }
152 }
153
154 //method for showing the first sheep from the list
155 void Animal::DisplayFirstSheep()
156 {
157     if(head != NULL)
158     {
159         cout << left;
160         cout << setw(15) << "NAME" << "AGE" << endl;
161         cout << setw(15) << "-----" << "----" << endl;
162         cout << setw(16) << nameAr[0] << ageAr[0];
163         cout << endl << endl << endl;
164
165         cout << "Is at the front of the list!" << endl << endl;

```

```

166     }
167
168     else if(head == NULL)
169     {
170         cout << "Nobody is in front -the list is empty!" << endl << endl;
171     }
172
173
174 }
175
176 //method for finding the sheep in the list
177 void Animal::FindSheep(string search) const
178 {
179     /*****
180      *   VARIABLE *
181      *****/
182
183     int    i;
184     bool   stat;
185
186     /*****
187      *   INITIALIZATION *
188      *****/
189
190     i      = 0;
191     stat = false;
192
193     if(head != NULL)
194     {
195         while(i < AR_SIZE && stat)
196         {
197             if(nameAr[i] == search)
198             {
199                 stat = true;
200             }
201
202             else
203             {
204                 i++;
205             }
206
207             if(stat == true)
208             {
209                 cout << setw(15) << "NAME" << "AGE" << endl;
210                 cout << setw(15) << "-----" << "----" << endl;
211                 cout << setw(16) << nameAr[i] << ageAr[i] << endl << endl;
212                 cout << "Has Been Found";
213             }
214         }
215     }
216
217     else if(head == NULL)
218     {
219         cout << "There are no sheep to be found!" << endl << endl;
220     }

```

```

221
222 }
223
224
225 //method for outputting the objects
226 void Animal::Display() const
227 {
228     /*****
229      *   VARIABLE *
230      *****/
231
232     SheepNode *sheepPtr;
233
234     if(head != NULL)
235     {
236         cout << "<output using the array>" << endl;
237         cout << left;
238         cout << setw(15) << "NAME" << "AGE" << endl;
239         cout << setw(15) << "-----" << "----" << endl;
240
241         for(int index = 0; index < listSize; index++)
242         {
243             cout << setw(16) << nameAr[index] << ageAr[index] << endl;
244         }
245         cout << endl << endl;
246
247         cout << "<output using the linked list>" << endl;
248         cout << left;
249         cout << setw(15) << "NAME" << "AGE" << endl;
250         cout << setw(15) << "-----" << "----" << endl;
251
252         for(sheepPtr = head -> next; sheepPtr != NULL; sheepPtr = sheepPtr ->
253 next)
254         {
255             cout << setw(16) << sheepPtr->sheepName << sheepPtr->sheepAge << endl;
256         }
257     }
258
259     else if(head == NULL)
260     {
261         cout << "\nThe list is empty" << endl << endl;
262     }
263
264 }
265 }
266
267
268
269

```