```
**************************************************
*  PROGRAMMED BY : ALI ESHGHI
*  CLASS         : CS1B
*  SECTION       : MW: 7:30p - 9:50p
*  LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
**************************************************

************************************
* WELCOME TO THE SHEEP LIST MANAGER *
************************************

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 1
Sheep name: Fluffy
Sheep Age:  1


The Sheep...
Sheep Name: Fluffy
Sheep Age:  1
Has been added

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 2
NAME           AGE
-------------- ---
Fluffy          1


Is at the front of the list!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 1
Sheep name: Maa
```

```
Sheep Age:  3


The Sheep...
Sheep Name: Maa
Sheep Age:  3
Has been added

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 4
There are 2 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 5
<output using the array>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3


<output using the linked list>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3
There are 2 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 1
Sheep name: Baa Baa
Sheep Age:  2


```

```
The Sheep...
Sheep Name: Baa Baa
Sheep Age:  2
Has been added

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 5
<output using the array>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3
Baa Baa         2


<output using the linked list>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3
Baa Baa         2
There are 3 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 4
There are 3 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 3
Who are you looking for? Baa Baa

NAME           AGE
-------------- ---

```

```
Baa Baa         2

Has Been Found!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 6

The list has been cleared!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 6

The list is empty

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 5

The list is empty


SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 4

The list is empty


SHEEP LIST MANAGER
```

```
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 3

There are no sheep to be found!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 2
Nobody is in front -the list is empty!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 7

**** The number 7 is an invalid entry     ****
**** Please input a number between 0 or 6 ****

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 0


```

```
 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
```

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

```cpp
/*************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/
#include "MyHeader.h"
#include "Classheader.h"

/*************************************************************
 * LAB 13 - ARRAYS AND LINKED LIST (OOP)
 *_____________________________________________________________
 * This program will use the arrays and linked list as an
 * objects of a class and gets the information base of the
 * users choice and add those information to the list and
 * prompts a menu for the user who can have more options
 *_____________________________________________________________
 * INPUT:menuOption -> user input for the menu
 *       name       -> name of a new sheep
 *       age    -> age of a new sheep
 *
 *
 * OUTPUT:name      -> name of the sheeps in the list
 *        age       -> age of the sheeps in the list
 *        size      -> how many sheeps are in the list
 *        search-> user input name to search in the list
 *        if the list is empty or no
 *************************************************************/

int main()
{

    /*************
     *  VARIABLE *
     *************/
    int     menuOption;
    Animal  sheep;
    string  name;
    string  search;
    int     age;
    int     size;


    //this function will output the class header
    PrintHeader();



    cout << "*************************************" << endl;
    cout << "* WELCOME TO THE SHEEP LIST MANAGER *" << endl;
    cout << "*************************************" << endl << endl;

    //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         {
```

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

```cpp
/*************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/

#include"MyHeader.h"

/*************************************************************
 * Function - Menu
 * ----------------------------------------------------------
 * This function will outputs the menu and prompts the user
 * to choose an option from the menu
 *
 * return type - integer
 *               the function is int type
 *************************************************************/

int Menu()
{

    /*************
     *  VARIABLE *
     *************/

    int menuOption;
    bool checkInp;

    /*******************
     *  INITIALIZATION *
     *******************/

    checkInp = false;

    //do while loop for input check
    do
    {

    cout << "SHEEP LIST MANAGER" << endl;
    cout << "1 - Add Sheep" << endl;
    cout << "2 - Output 1st Sheep" << endl;
    cout << "3 - Find Sheep" << endl;
    cout << "4 - List Size" << endl;
    cout << "5 - Output List" << endl;
    cout << "6 - Clear list" << endl;
    cout << "0 - Exit" << endl;
    cout << "Enter a command: ";



    //CHECKS FOR THE CHAR INPUT

```

```
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
```

```cpp
/*************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/

#include "MyHeader.h"

/*************************************************************
 * Function - PrintHeaderFile
 * ----------------------------------------------------------
 * This function will output the class heading to the screen.
 *
 * return type - nothing
 *               the function is void type
 *************************************************************/

void PrintHeader()
{
    cout << left;
    cout << "*******************************************************\n"       ;
    cout << "*  PROGRAMMED BY : ALI ESHGHI"                                ;
    cout << "\n*  "     << setw(14) << "CLASS"     << ": " << CLASS       ;
    cout << "\n*  "     << setw(14) << "SECTION"   << ": " << SECTION     ;
    cout << "\n*  LAB #"<< setw(9)  << NUM  << ": " << NAME           ;
    cout << "\n*******************************************************\n\n"   ;
    cout << right;
}


```

```
/*************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/

#ifndef CLASSHEADER_H_
#define CLASSHEADER_H_

#include<iostream>
#include<iomanip>
#include<string>
#include<fstream>
#include<limits>
#include<sstream>
using namespace std;

const int AR_SIZE = 50;


class Animal
{
    //public part of the class that is available for outside of the class
    public:
        //constructor
        Animal();

        //decosntructor
        ~Animal();

        //method for adding a new sheep and its age to parallel arrays
        void AddSheep(string name, int age);

        //method for adding a new sheep to a linked list
        void AddSheepLinkedList(string name, int age);

        //method for showing the first sheep from the list
        void DisplayFirstSheep();

        //method that returns the size of the list of the sheeps
        int ListSize() const;

        //method  for outputting the objects
        void Display() const;

        //method for finding the sheep in the list
        void FindSheep(string) const;


    //private part only available for the class
    private:
        string nameAr[AR_SIZE];
```

```
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
```

```cpp
/*************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/

#include "MyHeader.h"
#include "ClassHeader.h"


Animal::Animal()/*** CONSTRUCTOR ***/
{
    /*******************
     *  INITIALIZATION *
     *******************/

    name.clear();
    age  = 0;
    listSize = 0;
    head = NULL;
}

Animal::~Animal()    /*** DESTRUCTOR ***/
{

    /*************
     *  VARIABLE *
     *************/

    SheepNode *sheepPtr;

    if(head != NULL)
    {
        //clear the list
        sheepPtr = head;
        while(sheepPtr != NULL)
        {
            head = head -> next;
            delete sheepPtr;

            sheepPtr = head;
        }

        for(int i = 0; i < AR_SIZE; i++)
        {
            nameAr[i] = ' ';
        }
        cout << "The list has been cleared!" << endl << endl;
    }

    else if(head == NULL)
    {
```

```
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
248         cout << "<output using the linked list>" << endl;
249         cout << left;
250         cout << setw(15) << "NAME" << "AGE" <<endl;
251         cout << setw(15) << "---------------" << "---" << endl;
252
253         for(sheepPtr = head -> next; sheepPtr != NULL; sheepPtr = sheepPtr ->
    next)
254         {
255             cout << setw(16) << sheepPtr->sheepName << sheepPtr->sheepAge << endl;
256         }
257
258     }
259
260     else if(head == NULL)
261     {
262         cout << "\nThe list is empty" << endl << endl;
263     }
264
265 }
266
267
268
269
```