```
 1 *********************************************
 2 * PROGRAMMED BY : Faris Hijazi
 3 * CLASS         : CS1A
 4 * SECTION       : MW: 7:30P
 5 * Assignment #13: Arrays and Linked Lists of Sheep - Array
 6 *********************************************
 7
 8 1 - Add Sheep
 9 2 - Output first sheep
10 3 - Find sheep
11 4 - List size
12 5 - Output list
13 6 - Clear list
14 0 - Exit
15 enter a command ? 1
16
17 Sheep Name: Fluffy
18 Sheep Age:  1
19
20 The sheep...
21 Sheep Name: Fluffy
22 Sheep Age:  1
23 Has been added
24
25 1 - Add Sheep
26 2 - Output first sheep
27 3 - Find sheep
28 4 - List size
29 5 - Output list
30 6 - Clear list
31 0 - Exit
32 enter a command ? 2
33
34 NAME          AGE
35 ------------- ---
36 Fluffy        1
37
38 1 - Add Sheep
39 2 - Output first sheep
40 3 - Find sheep
41 4 - List size
42 5 - Output list
43 6 - Clear list
44 0 - Exit
45 enter a command ? 1
46
47 Sheep Name: Maa
48 Sheep Age:  3
49
50 The sheep...
51 Sheep Name: Maa
52 Sheep Age:  3
53 Has been added
54
55 1 - Add Sheep
56 2 - Output first sheep
57 3 - Find sheep
58 4 - List size
59 5 - Output list
60 6 - Clear list
61 0 - Exit
62 enter a command ? 4
63
64 There are 2 sheep in the list
```

```
65
1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 5

NAME          AGE
------------- ---
Fluffy        1
Maa           3

1 - Add Sheep
2 - Output first 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

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 5

NAME          AGE
------------- ---
Fluffy        1
Maa           3
Baa Baa       2

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 4

There are 3 sheep in the list

1 - Add Sheep
2 - Output first 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

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 6

the list has been cleared!

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 6

the list has been cleared!

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 5

no sheep

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 4

There are 0 sheep in the list

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 3

```

```
no sheep to search

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 2

no sheep in list

1 - Add Sheep
2 - Output first 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 and 6 ****

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 0

```

```
 1 /*****************************************************************************
 2  * AUTHOR       : Faris Hijazi
 3  * STUDENT ID   : 1039438
 4  * Lab 14       : Arrays and Linked Lists of Sheep
 5  * CLASS        : CS1A
 6  * SECTION      : MW 7:30PM
 7  * DUE DATE     : 05/09/19
 8  *****************************************************************************/
 9
10 #ifndef HEADER_H_
11 #define HEADER_H_
12 #include <string>
13 #include <iostream>
14 #include <iomanip>
15 #include <string>
16 #include <limits>
17 #include <ios>
18 using namespace std;
19
20 enum menu
21 {
22     EXIT,
23     ADD,
24     PEEK,
25     SEARCH,
26     SIZE,
27     OUTPUT,
28     CLEAR
29 };
30
31 void outputMenu();
32
33 int menuInput();
34
35 void PrintHeader(ostream &output,   //output device to use
36                  char exersize,     //assignment or lab
37                  string exersizeName,//name of exersize
38                  int num,           //as or lab number
39                  string names);     //names
40
41 #endif /* HEADER_H_ */
```

```cpp
/*******************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * Lab 13       : Arrays and Linked Lists of Sheep
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 05/09/19
 *******************************************************************************/

#include "sheepHeaderArray.h"

int main()
{
    const int NCOL_SIZE = 14;
    const int ACOL_SIZE = 3;

    int sheepAge;
    string sheepName;
    bool invalid;
    Farm newFarm;
    Sheep newSheep;
    int menuOpt;

    PrintHeader(cout, 'l',"Arrays and Linked Lists of Sheep - Array", 13, "Faris Hijazi");

    menuOpt = menuInput();
    while(menuOpt!=0)
    {
        switch (menuOpt)
        {
            case ADD:
                cout << endl;
                cout << left << setw(12) << "Sheep Name:";
                getline(cin, sheepName);
                //error check sheep age
                do
                {
                    cout <<setw(12) << "Sheep Age:";
                    if(!(cin >> sheepAge))
                        {
                            cout << "\n**** Please input a number between 0 and 9 ****\n";
                            cin.clear();
                            cin.ignore(numeric_limits<streamsize>::max(), '\n');
                            invalid = true;
                        }
                        else if(sheepAge < 0 || sheepAge > 9)
                        {
                            cout << "\n**** The number " <<  sheepAge << " is an invalid entry
****\n";
                            cout << "**** Please input a number between 0 and 9 ****\n";
                            invalid =  true;
                        }
                        else
                        {
                            cin.ignore(1000, '\n');
                            invalid = false;
                        }
                    cout << endl;
                }while(invalid);
                newSheep.SetInitialValues(sheepName, sheepAge);
                newFarm.AddSheep(newSheep);
                break;

            case PEEK:
```

```
64                  if(newFarm.TotalSheep() > 0)
65                  {
66                      cout << endl;
67                      cout << left;
68                      cout << setw(NCOL_SIZE) << "NAME";
69                      cout << setw(NCOL_SIZE) << "AGE";
70                      cout << endl;
71                      cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
72                      cout << ' ';
73                      cout << setfill('-') << setw(ACOL_SIZE) << '-';
74                      cout << setfill(' ');
75                      cout << endl;
76                      newFarm.GetFirstSheep().GetValues(sheepName, sheepAge);
77                      cout << setw(NCOL_SIZE) << sheepName;
78                      cout << setw(ACOL_SIZE) << sheepAge;
79                      cout << endl;
80                  }
81                  else
82                  {
83                      cout << "\nno sheep in list\n";
84                  }
85
86                  break;
87              case SEARCH:
88                  if (newFarm.TotalSheep() > 0)
89                  {
90                      cout << "\nwho are you looking for? ";
91                      getline(cin, sheepName);
92
93                      cout << endl;
94                      cout << left;
95                      cout << setw(NCOL_SIZE) << "NAME";
96                      cout << setw(NCOL_SIZE) << "AGE";
97                      cout << endl;
98                      cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
99                      cout << ' ';
100                     cout << setfill('-') << setw(ACOL_SIZE) << '-';
101                     cout << setfill(' ');
102                     cout << endl;
103                     newFarm.FindSheep(sheepName).GetValues(sheepName, sheepAge);
104                     cout << setw(NCOL_SIZE) << sheepName;
105                     cout << setw(ACOL_SIZE) << sheepAge;
106                     cout << endl <<endl;
107
108                     cout << "has been found\n";
109                 }
110                 else
111                 {
112                     cout << "\nno sheep to search\n";
113                 }
114                 break;
115             case SIZE:
116                 cout <<endl;
117                 cout << "There are " << newFarm.TotalSheep()
118                     << " sheep in the list\n";
119                 break;
120             case OUTPUT:
121                 cout << endl;
122                 newFarm.DisplaySheepTable();
123                 break;
124             case CLEAR:
125                 newFarm.ClearList();
126                 cout << "\nthe list has been cleared!\n";
127                 break;
```

```
128         }
129         menuOpt = menuInput();
130     }
131         return 0;
132 }
```

```cpp
/*****************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * Lab 14       : Arrays and Linked Lists of Sheep
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 05/09/19
 *****************************************************************************/

/*****************************************************************************
 * This function will output the menu to the console
 *----------------------------------------------------------------------------
 * INPUT:
 *          NA
 * OUTPUT:
 *          NA
 *****************************************************************************/

#include "sheepHeaderArray.h"

void outputMenu()
{
    cout << "\n1 - Add Sheep\n";
    cout << "2 - Output first sheep\n";
    cout << "3 - Find sheep\n";
    cout << "4 - List size\n";
    cout << "5 - Output list\n";
    cout << "6 - Clear list\n";
    cout << "0 - Exit\n";
    cout << "enter a command ? ";
}

/*****************************************************************************
 * This function will get and check menu input
 *----------------------------------------------------------------------------
 * INPUT:
 *          NA
 * OUTPUT:
 *          NA
 *****************************************************************************/

int menuInput()
{
    int menuOpt;
    bool invalid = false;

    do
    {
        outputMenu();
        if(!(cin >> menuOpt))
        {
            cout << "\n**** Please input a number between 0 and 6 ****\n";
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            invalid = true;
        }
        else if(menuOpt < 0 || menuOpt > 6)
        {
            cout << "\n**** The number " <<  menuOpt << " is an invalid entry ****\n";
            cout << "**** Please input a number between 0 and 6 ****\n";
            invalid =  true;
        }
        else
        {
            cin.ignore(1000, '\n');
```

```
65              invalid = false;
66          }
67
68      }while(invalid);
69
70      return menuOpt;
71  }
72
73  /*****************************************************************************
74   * This function will output the class header using ostream
75   *----------------------------------------------------------------------------
76   * INPUT:
77   *      output      - output file variable
78   *      exersize    - Lab or Assignment
79   *      exersizeName- name of exersize
80   *      num         - number of Lab/Assignment
81   *      names       - names of programmers
82   * OUTPUT:
83   *      header
84   *****************************************************************************/
85  void PrintHeader(ostream &output, char exersize, string exersizeName,  int num, string names)
86  {
87
88      int colWidth;  //CALC - changes based on exersize
89      string asType; //CALC - changes based on exersize
90
91      if(exersize == 'L')
92      {
93          asType = "Lab";
94          colWidth = 9;
95      }
96      else
97      {
98          asType = "Assignment";
99          colWidth = 2;
100     }
101
102     output << left;
103     output <<"*********************************************\n";
104     output <<"* PROGRAMMED BY : "  << names << endl;
105     output <<"* "<< setw(14) << "CLASS"   << ": " << "CS1A"    << endl;
106     output <<"* "<< setw(14) << "SECTION" << ": " << "MW: 7:30P"  << endl;
107     output <<"* "<< asType << " #" << setw(colWidth)  << num   << ": " << exersizeName << endl;
108     output <<"*********************************************\n";
109     output << right;
110 }
111
```

```
1 /*****************************************************************************
2  * AUTHOR       : Faris Hijazi
3  * STUDENT ID   : 1039438
4  * Lab 14       : Arrays and Linked Lists of Sheep
5  * CLASS        : CS1A
6  * SECTION      : MW 7:30PM
7  * DUE DATE     : 05/09/19
8  *****************************************************************************/
9
10 #ifndef HEADER_H_ARRAY
11 #define HEADER_H_ARRAY
12 #include "mainheader.h"
13
14 const int AR_SIZE = 50;
15
16 class Sheep
17 {
18     public:
19         Sheep();
20         ~Sheep();
21         /******************
22          ***  MUTATORS  ***
23          ******************/
24         void SetInitialValues(string sheepName,int sheepAge);//sets values for age amd name
25         /*******************
26          ***  ACCESSORS  ***
27          *******************/
28         void GetValues(string &sheepName,int &sheepAge) const;//gets values for age and name
29         string GetName() const;//returns name as string
30     private:
31         string name;
32         int age;
33 };
34
35 class Farm
36 {
37     public:
38         Farm();
39         ~Farm();
40         /******************
41          ***  MUTATORS  ***
42          ******************/
43         //add new sheep object to the list incriment sheep count
44         void AddSheep(Sheep newSheep);//Adds new sheep to list
45         void ClearList();//remove all sheep
46         /******************
47          *** ACCESSORS  ***
48          ******************/
49         Sheep FindSheep(string sheepName) const;//Find sheep and return the object
50         Sheep GetFirstSheep() const;//Return first sheep
51         int TotalSheep() const;//Displays number of sheep
52         void DisplaySheepTable() const;//Output all sheep
53
54     private:
55     Sheep sheepArray[AR_SIZE];//list of sheep objects
56     int sheepCount;//number of sheep in list
57 };
58
59 #endif /* HEADER_H_ARRAY */
```

```cpp
#include "sheepHeaderArray.h"
//sheep methods

/*****************************************************************************
 * CONSTRUCTOR Animal
 *----------------------------------------------------------------------------
 *
 * RETURN: NA
 *****************************************************************************/
Sheep ::Sheep()
{
}
//---------------------------------------------------------------------------

/*****************************************************************************
 * DECONSTRUCTOR Animal
 *----------------------------------------------------------------------------
 *
 * RETURN: NA
 *****************************************************************************/
Sheep ::~Sheep()
{
}
//---------------------------------------------------------------------------

/*****************************************************************************
 * METHOD SetInitialValues
 *----------------------------------------------------------------------------
 * sets values for sheepName and SheepAge
 * RETURN: NA
 *****************************************************************************/
void Sheep::SetInitialValues(string sheepName,int sheepAge)
{
    name = sheepName;
    age = sheepAge;

    cout << "The sheep...\n";
    cout << setw(12) << "Sheep Name:" << name << endl;
    cout <<setw(12) << "Sheep Age:" << age << endl;
    cout << "Has been added\n";
}
//---------------------------------------------------------------------------

/*****************************************************************************
 * METHOD GetValues
 *----------------------------------------------------------------------------
 * gets values of sheepName and sheepAge
 * RETURN: NA
 *****************************************************************************/
void Sheep::GetValues(string &sheepName,int &sheepAge) const
{
    sheepAge = age;
    sheepName = name;
}
//---------------------------------------------------------------------------

/*****************************************************************************
 * METHOD GetName
 *----------------------------------------------------------------------------
 * Returns name
 * RETURN: name
 *****************************************************************************/
string Sheep::GetName() const
{
```

```
65      return name;
66  }
67
68  //Farm methods
69
70  /*****************************************************************************
71   * Farm CONSTRUCTOR
72   *----------------------------------------------------------------------------
73   * RETURN: NA
74   *****************************************************************************/
75  Farm ::Farm()
76  {
77      sheepCount = 0;
78  }
79  //----------------------------------------------------------------------------
80
81  /*****************************************************************************
82   * Farm DECONSTRUCTOR
83   *----------------------------------------------------------------------------
84   * RETURN: NA
85   *****************************************************************************/
86  Farm ::~Farm()
87  {
88  }
89  //----------------------------------------------------------------------------
90
91  /*****************************************************************************
92   * METHOD AddSheep
93   *----------------------------------------------------------------------------
94   * adds a sheep to the the list
95   * RETURN: NA
96   *****************************************************************************/
97  void Farm::AddSheep(Sheep newSheep)
98  {
99      if(sheepCount-1 < AR_SIZE)
100     {
101         sheepArray[sheepCount] = newSheep;
102         sheepCount++;
103     }
104 }
105 //----------------------------------------------------------------------------
106
107 /*****************************************************************************
108  * METHOD ClearList
109  *----------------------------------------------------------------------------
110  * clears the list of all sheep
111  * RETURN: NA
112  *****************************************************************************/
113 void Farm::ClearList()
114 {
115     sheepCount = 0;
116 }
117 //----------------------------------------------------------------------------
118
119 /*****************************************************************************
120  * METHOD FindSheep
121  *----------------------------------------------------------------------------
122  * Finds sheep with specified name
123  * RETURN: sheep to search
124  *****************************************************************************/
125 Sheep Farm::FindSheep(string sheepName) const
126 {
127     int i = 0;
128     bool found = false;
```

```
129
130     while(i < sheepCount-1 && !found)
131     {
132         if(sheepName == sheepArray[i].GetName())
133         {
134             found = true;
135         }
136         else
137         {
138             i++;
139         }
140     }
141     return(sheepArray[i]);
142
143 }
144 //-------------------------------------------------------------------------
145
146 /*************************************************************************
147  * METHOD GetFirstSheep
148  *-----------------------------------------------------------------------
149  * returns the first sheep
150  * RETURN: first sheep
151  ************************************************************************/
152 Sheep Farm::GetFirstSheep() const
153 {
154         return(sheepArray[0]);
155
156 }
157 //-------------------------------------------------------------------------
158
159 /*************************************************************************
160  * METHOD TotalSheep
161  *-----------------------------------------------------------------------
162  * returns total sheep
163  * RETURN: sheepCount
164  ************************************************************************/
165 int Farm::TotalSheep() const
166 {
167     return(sheepCount);
168 }
169 //-------------------------------------------------------------------------
170
171 /*************************************************************************
172  * METHOD DisplaySheepTable
173  *-----------------------------------------------------------------------
174  * displays all sheep
175  * RETURN: NA
176  ************************************************************************/
177 void Farm::DisplaySheepTable() const
178 {
179     if(sheepCount > 0)
180     {
181         const int NCOL_SIZE = 14;
182         const int ACOL_SIZE = 3;
183         int i = 0;
184         int sheepAge;
185         string sheepName;
186
187         cout << left;
188         cout << setw(NCOL_SIZE) << "NAME";
189         cout << setw(NCOL_SIZE) << "AGE";
190         cout << endl;
191         cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
192         cout << ' ';
```

```
193         cout << setfill('-') << setw(ACOL_SIZE) << '-';
194         cout << setfill(' ');
195         cout << endl;
196         while(i < sheepCount)
197         {
198             sheepArray[i].GetValues(sheepName, sheepAge);
199             cout << setw(NCOL_SIZE) << sheepName;
200             cout << setw(ACOL_SIZE) << sheepAge;
201             cout << endl;
202 
203             i++;
204         }
205     }
206     else
207     {
208         cout << "no sheep\n";
209     }
210 }
211 
212 
```

```
 1 *********************************************
 2 * PROGRAMMED BY : Faris Hijazi
 3 * CLASS         : CS1A
 4 * SECTION       : MW: 7:30P
 5 * Assignment #13: Arrays and Linked Lists of Sheep - LL
 6 *********************************************
 7
 8 1 - Add Sheep
 9 2 - Output first sheep
10 3 - Find sheep
11 4 - List size
12 5 - Output list
13 6 - Clear list
14 0 - Exit
15 enter a command ? 1
16
17 Sheep Name: Fluffy
18 Sheep Age:  1
19
20 The sheep...
21 Sheep Name: Fluffy
22 Sheep Age:  1
23 Has been added
24
25 1 - Add Sheep
26 2 - Output first sheep
27 3 - Find sheep
28 4 - List size
29 5 - Output list
30 6 - Clear list
31 0 - Exit
32 enter a command ? 2
33
34 NAME          AGE
35 ------------- ---
36 Fluffy        1
37
38 1 - Add Sheep
39 2 - Output first sheep
40 3 - Find sheep
41 4 - List size
42 5 - Output list
43 6 - Clear list
44 0 - Exit
45 enter a command ? 1
46
47 Sheep Name: Maa
48 Sheep Age:  3
49
50 The sheep...
51 Sheep Name: Maa
52 Sheep Age:  3
53 Has been added
54
55 1 - Add Sheep
56 2 - Output first sheep
57 3 - Find sheep
58 4 - List size
59 5 - Output list
60 6 - Clear list
61 0 - Exit
62 enter a command ? 4
63
64 There are 2 sheep in the list
```

```
 65
 66 1 - Add Sheep
 67 2 - Output first sheep
 68 3 - Find sheep
 69 4 - List size
 70 5 - Output list
 71 6 - Clear list
 72 0 - Exit
 73 enter a command ? 5
 74
 75 NAME          AGE
 76 ------------- ---
 77 Fluffy        1
 78 Maa           3
 79
 80 1 - Add Sheep
 81 2 - Output first sheep
 82 3 - Find sheep
 83 4 - List size
 84 5 - Output list
 85 6 - Clear list
 86 0 - Exit
 87 enter a command ? 1
 88
 89 Sheep Name: Baa Baa
 90 Sheep Age:  2
 91
 92 The sheep...
 93 Sheep Name: Baa Baa
 94 Sheep Age:  2
 95 Has been added
 96
 97 1 - Add Sheep
 98 2 - Output first sheep
 99 3 - Find sheep
100 4 - List size
101 5 - Output list
102 6 - Clear list
103 0 - Exit
104 enter a command ? 5
105
106 NAME          AGE
107 ------------- ---
108 Fluffy        1
109 Maa           3
110 Baa Baa       2
111
112 1 - Add Sheep
113 2 - Output first sheep
114 3 - Find sheep
115 4 - List size
116 5 - Output list
117 6 - Clear list
118 0 - Exit
119 enter a command ? 4
120
121 There are 3 sheep in the list
122
123 1 - Add Sheep
124 2 - Output first sheep
125 3 - Find sheep
126 4 - List size
127 5 - Output list
128 6 - Clear list
```

```
0 - Exit
enter a command ? 3

who are you looking for? Baa Baa

NAME          AGE
------------- ---
Baa Baa       2

has been found

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 6

the list has been cleared!

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 6

the list has been cleared!

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 5

no sheep

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 4

There are 0 sheep in the list

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 3

```

```
no sheep to search

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 2

no sheep in list

1 - Add Sheep
2 - Output first 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 and 6 ****

1 - Add Sheep
2 - Output first sheep
3 - Find sheep
4 - List size
5 - Output list
6 - Clear list
0 - Exit
enter a command ? 0

```

```
/*****************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * Lab 14       : Arrays and Linked Lists of Sheep
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 05/09/19
 *****************************************************************************/

#ifndef HEADER_H_LL
#define HEADER_H_LL
#include "mainheader.h"

const int AR_SIZE = 50;

class Sheep
{
    public:
        Sheep();
        ~Sheep();
        /******************
         ***  MUTATORS  ***
         ******************/
        void SetInitialValues(string sheepName,int sheepAge);
        /*******************
         ***  ACCESSORS  ***
         *******************/
        void GetValues(string &sheepName,int &sheepAge) const;
        string GetName() const;
    private:
        string name;
        int age;
};

class Farm
{
    public:
        Farm();
        ~Farm();
        /******************
         ***  MUTATORS  ***
         ******************/
        //add new sheep object to the list incriment sheep count
        void AddSheep(Sheep newSheep);//Adds new sheep to list
        void ClearList();//remove all sheep
        /******************
         *** ACCESSORS  ***
         ******************/
        Sheep FindSheep(string sheepName) const;
        Sheep GetFirstSheep() const;//Return first sheep
        int TotalSheep() const;//Displays number of sheep
        void DisplaySheepTable() const;//Output all sheep

    private:

    struct SheepNode
    {
        Sheep currentSheep; //sheep object
        SheepNode *next;    //next pointer
    };
    SheepNode *head;//head pointer
    int sheepCount;//number of sheep in LL
};

```

```
65 #endif /* HEADER_H_LL */
```

```cpp
/*****************************************************************************
 * AUTHOR       : Faris Hijazi
 * STUDENT ID   : 1039438
 * Lab 14       : Arrays and Linked Lists of Sheep
 * CLASS        : CS1A
 * SECTION      : MW 7:30PM
 * DUE DATE     : 05/09/19
 *****************************************************************************/

#include "sheepHeaderLL.h"
//sheep methods

/*****************************************************************************
 * CONSTRUCTOR Animal
 *----------------------------------------------------------------------------
 *
 * RETURN: NA
 *****************************************************************************/
Sheep ::Sheep()
{
}
//----------------------------------------------------------------------------

/*****************************************************************************
 * DECONSTRUCTOR Animal
 *----------------------------------------------------------------------------
 *
 * RETURN: NA
 *****************************************************************************/
Sheep ::~Sheep()
{
}
//----------------------------------------------------------------------------

/*****************************************************************************
 * METHOD SetInitialValues
 *----------------------------------------------------------------------------
 * sets values for sheepName and SheepAge
 * RETURN: NA
 *****************************************************************************/
void Sheep::SetInitialValues(string sheepName,int sheepAge)
{
    name = sheepName;
    age = sheepAge;

    cout << "The sheep...\n";
    cout << setw(12) << "Sheep Name:" << name << endl;
    cout <<setw(12) << "Sheep Age:" << age << endl;
    cout << "Has been added\n";
}
//----------------------------------------------------------------------------

/*****************************************************************************
 * METHOD GetValues
 *----------------------------------------------------------------------------
 * gets values of sheepName and sheepAge
 * RETURN: NA
 *****************************************************************************/
void Sheep::GetValues(string &sheepName,int &sheepAge) const
{
    sheepAge = age;
    sheepName = name;
}
//----------------------------------------------------------------------------
```

```
 65
 66 /*****************************************************************************
 67  * METHOD GetName
 68  *----------------------------------------------------------------------------
 69  * Returns name
 70  * RETURN: name
 71  *****************************************************************************/
 72 string Sheep::GetName() const
 73 {
 74     return name;
 75 }
 76
 77 //Farm methods
 78
 79 /*****************************************************************************
 80  * Farm CONSTRUCTOR
 81  *----------------------------------------------------------------------------
 82  * RETURN: NA
 83  *****************************************************************************/
 84 Farm ::Farm()
 85 {
 86     head = NULL;
 87     sheepCount = 0;
 88 }
 89 //----------------------------------------------------------------------------
 90
 91 /*****************************************************************************
 92  * Farm DECONSTRUCTOR
 93  *----------------------------------------------------------------------------
 94  * RETURN: NA
 95  *****************************************************************************/
 96 Farm ::~Farm()
 97 {
 98     SheepNode *sheepPtr;
 99
100     sheepCount = 0;
101     while(head != NULL)
102     {
103         sheepPtr = head;
104         head = sheepPtr->next;
105         delete sheepPtr;
106     }
107     sheepPtr = NULL;
108 }
109 //----------------------------------------------------------------------------
110
111 /*****************************************************************************
112  * METHOD AddSheep
113  *----------------------------------------------------------------------------
114  * adds a sheep to the linked list
115  * RETURN: NA
116  *****************************************************************************/
117 void Farm::AddSheep(Sheep newSheep)
118 {
119     SheepNode *sheepPtr;
120     SheepNode *tail;
121
122     sheepCount++;
123     sheepPtr = new SheepNode;
124     if(head!=NULL)
125     {
126         tail = head;
127         while(tail->next != NULL)
128         {
```

```
129             tail = tail->next;
130         }
131         sheepPtr->next = NULL;
132         tail->next = sheepPtr;
133     }
134     else
135     {
136         sheepPtr->next = head;
137         head = sheepPtr;
138     }
139     sheepPtr->currentSheep = newSheep;
140     sheepPtr = NULL;
141 }
142 //--------------------------------------------------------------------------
143
144 /*****************************************************************************
145  * METHOD ClearList
146  *---------------------------------------------------------------------------
147  * clears the list of all sheep
148  * RETURN: NA
149  ****************************************************************************/
150 void Farm::ClearList()
151 {
152     SheepNode *sheepPtr;
153
154     sheepCount = 0;
155     while(head != NULL)
156     {
157         sheepPtr = head;
158         head = sheepPtr->next;
159         delete sheepPtr;
160     }
161     sheepPtr = NULL;
162 }
163 //--------------------------------------------------------------------------
164
165 /*****************************************************************************
166  * METHOD FindSheep
167  *---------------------------------------------------------------------------
168  * Finds sheep with specified name
169  * RETURN: sheep to search
170  ****************************************************************************/
171 Sheep Farm::FindSheep(string sheepName) const
172 {
173     int i = 0;
174     bool found = false;
175     SheepNode *sheepPtr;
176     if(sheepCount > 0)
177     {
178         sheepPtr = head;
179         while(sheepPtr!=NULL && !found)
180         {
181             if (sheepPtr->currentSheep.GetName() == sheepName)
182             {
183                 found = true;
184             }
185             else
186             {
187                 sheepPtr = sheepPtr->next;
188             }
189         }
190     }
191     return(sheepPtr->currentSheep);
192 }
```

```
193 //---------------------------------------------------------------------------
194
195 /*****************************************************************************
196  * METHOD GetFirstSheep
197  *---------------------------------------------------------------------------
198  * returns the first sheep
199  * RETURN: first sheep
200  *****************************************************************************/
201 Sheep Farm::GetFirstSheep() const
202 {
203     return(head->currentSheep);
204 }
205 //---------------------------------------------------------------------------
206
207 /*****************************************************************************
208  * METHOD TotalSheep
209  *---------------------------------------------------------------------------
210  * returns total sheep
211  * RETURN: sheepCount
212  *****************************************************************************/
213 int Farm::TotalSheep() const
214 {
215     return(sheepCount);
216 }
217 //---------------------------------------------------------------------------
218
219 /*****************************************************************************
220  * METHOD DisplaySheepTable
221  *---------------------------------------------------------------------------
222  * displays all sheep
223  * RETURN: NA
224  *****************************************************************************/
225 void Farm::DisplaySheepTable() const
226 {
227     if(sheepCount > 0)
228     {
229         const int NCOL_SIZE = 14;
230         const int ACOL_SIZE = 3;
231         int i = 0;
232         int sheepAge;
233         string sheepName;
234         SheepNode *sheepPtr;
235
236         cout << left;
237         cout << setw(NCOL_SIZE) << "NAME";
238         cout << setw(NCOL_SIZE) << "AGE";
239         cout << endl;
240         cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
241         cout << ' ';
242         cout << setfill('-') << setw(ACOL_SIZE) << '-';
243         cout << setfill(' ');
244         cout << endl;
245
246         sheepPtr = head;
247         while(sheepPtr!=NULL)
248         {
249             sheepPtr->currentSheep.GetValues(sheepName, sheepAge);
250             cout << setw(NCOL_SIZE) << sheepName;
251             cout << setw(ACOL_SIZE) << sheepAge;
252             cout << endl;
253
254             sheepPtr = sheepPtr->next;
255         }
256     }
```

```
257     else
258     {
259         cout << "no sheep\n";
260     }
261 }
```