```
************************************************
* PROGRAMMED BY : Faris Hijazi
* CLASS         : CS1A
* SECTION       : MW: 7:30P
* Assignment #13: Arrays and Linked Lists of Sheep - Array
************************************************

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: Fluffy
Sheep Age:  1

The sheep...
Sheep Name: Fluffy
Sheep Age:  1
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 ? 2

NAME          AGE
------------- ---
Fluffy        1

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: Maa
Sheep Age:  3

The sheep...
Sheep Name: Maa
Sheep Age:  3
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 ? 4

There are 2 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 ? 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

```

mainheader.h

```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
 *****************************************************************************/

#ifndef HEADER_H_
#define HEADER_H_
#include <string>
#include <iostream>
#include <iomanip>
#include <string>
#include <limits>
#include <ios>
using namespace std;

enum menu
{
    EXIT,
    ADD,
    PEEK,
    SEARCH,
    SIZE,
    OUTPUT,
    CLEAR
};

void outputMenu();

int menuInput();

void PrintHeader(ostream &output,   //output device to use
                char exersize,      //assignment or lab
                string exersizeName,//name of exersize
                int num,            //as or lab number
                string names);      //names

#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:
```

```cpp
            if(newFarm.TotalSheep() > 0)
            {
                cout << endl;
                cout << left;
                cout << setw(NCOL_SIZE) << "NAME";
                cout << setw(NCOL_SIZE) << "AGE";
                cout << endl;
                cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
                cout << ' ';
                cout << setfill('-') << setw(ACOL_SIZE) << '-';
                cout << setfill(' ');
                cout << endl;
                newFarm.GetFirstSheep().GetValues(sheepName, sheepAge);
                cout << setw(NCOL_SIZE) << sheepName;
                cout << setw(ACOL_SIZE) << sheepAge;
                cout << endl;
            }
            else
            {
                cout << "\nno sheep in list\n";
            }

            break;
        case SEARCH:
            if (newFarm.TotalSheep() > 0)
            {
                cout << "\nwho are you looking for? ";
                getline(cin, sheepName);

                cout << endl;
                cout << left;
                cout << setw(NCOL_SIZE) << "NAME";
                cout << setw(NCOL_SIZE) << "AGE";
                cout << endl;
                cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
                cout << ' ';
                cout << setfill('-') << setw(ACOL_SIZE) << '-';
                cout << setfill(' ');
                cout << endl;
                newFarm.FindSheep(sheepName).GetValues(sheepName, sheepAge);
                cout << setw(NCOL_SIZE) << sheepName;
                cout << setw(ACOL_SIZE) << sheepAge;
                cout << endl <<endl;

                cout << "has been found\n";
            }
            else
            {
                cout << "\nno sheep to search\n";
            }
            break;
        case SIZE:
            cout <<endl;
            cout << "There are " << newFarm.TotalSheep()
                << " sheep in the list\n";
            break;
        case OUTPUT:
            cout << endl;
            newFarm.DisplaySheepTable();
            break;
        case CLEAR:
            newFarm.ClearList();
            cout << "\nthe list has been cleared!\n";
            break;
        }
        menuOpt = menuInput();
    }
        return 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
 *****************************************************************************/

/******************************************************************************
 * 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');
            invalid = false;
        }

    }while(invalid);

    return menuOpt;
}

/******************************************************************************
 * This function will output the class header using ostream
 *----------------------------------------------------------------------------
 * INPUT:
 *      output      - output file variable
 *      exersize    - Lab or Assignment
 *      exersizeName- name of exersize
 *      num         - number of Lab/Assignment
 *      names       - names of programmers
 * OUTPUT:
 *      header
 *****************************************************************************/
void PrintHeader(ostream &output, char exersize, string exersizeName,  int num, string names)
{

    int colWidth;  //CALC - changes based on exersize
    string asType; //CALC - changes based on exersize

    if(exersize == 'L')
    {
        asType = "Lab";
        colWidth = 9;
    }
    else
    {
        asType = "Assignment";
        colWidth = 2;
    }

    output << left;
    output <<"************************************************\n";
    output <<"* PROGRAMMED BY : "  << names << endl;
    output <<"* "<< setw(14) << "CLASS"   << ": " << "CS1A"    << endl;
    output <<"* "<< setw(14) << "SECTION" << ": " << "MW: 7:30P"  << endl;
    output <<"* "<< asType << " #" << setw(colWidth)  << num   << ": " << exersizeName << endl;
    output <<"************************************************\n";
    output << right;
}

```

sheepHeaderArray.h

```
/******************************************************************************
 * 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_ARRAY
#define HEADER_H_ARRAY
#include "mainheader.h"

const int AR_SIZE = 50;

class Sheep
{
    public:
        Sheep();
        ~Sheep();
        /******************
         ***  MUTATORS  ***
         ******************/
        void SetInitialValues(string sheepName,int sheepAge);//sets values for age amd name
        /*******************
         ***  ACCESSORS  ***
         *******************/
        void GetValues(string &sheepName,int &sheepAge) const;//gets values for age and name
        string GetName() const;//returns name as string
    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;//Find sheep and return the object
        Sheep GetFirstSheep() const;//Return first sheep
        int TotalSheep() const;//Displays number of sheep
        void DisplaySheepTable() const;//Output all sheep

    private:
    Sheep sheepArray[AR_SIZE];//list of sheep objects
    int sheepCount;//number of sheep in list
};

#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
{
    return name;
}

//Farm methods

/*******************************************************************************
 * Farm CONSTRUCTOR
 *------------------------------------------------------------------------------
 * RETURN: NA
 ******************************************************************************/
Farm ::Farm()
{
    sheepCount = 0;
}
//------------------------------------------------------------------------------

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

/*******************************************************************************
 * METHOD AddSheep
 *------------------------------------------------------------------------------
 * adds a sheep to the the list
 * RETURN: NA
 ******************************************************************************/
void Farm::AddSheep(Sheep newSheep)
{
    if(sheepCount-1 < AR_SIZE)
    {
        sheepArray[sheepCount] = newSheep;
        sheepCount++;
    }
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD ClearList
 *------------------------------------------------------------------------------
 * clears the list of all sheep
 * RETURN: NA
 ******************************************************************************/
void Farm::ClearList()
{
    sheepCount = 0;
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD FindSheep
 *------------------------------------------------------------------------------
 * Finds sheep with specified name
 * RETURN: sheep to search
 ******************************************************************************/
Sheep Farm::FindSheep(string sheepName) const
{
    int i = 0;
    bool found = false;

    while(i < sheepCount-1 && !found)
    {
        if(sheepName == sheepArray[i].GetName())
        {
            found = true;
        }
        else
        {
            i++;
        }
    }
    return(sheepArray[i]);

}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD GetFirstSheep
 *------------------------------------------------------------------------------
 * returns the first sheep
 * RETURN: first sheep
 ******************************************************************************/
Sheep Farm::GetFirstSheep() const
{
        return(sheepArray[0]);

}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD TotalSheep
 *------------------------------------------------------------------------------
 * returns total sheep
 * RETURN: sheepCount
 ******************************************************************************/
int Farm::TotalSheep() const
{
    return(sheepCount);
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD DisplaySheepTable
 *------------------------------------------------------------------------------
 * displays all sheep
 * RETURN: NA
 ******************************************************************************/
void Farm::DisplaySheepTable() const
{
    if(sheepCount > 0)
    {
        const int NCOL_SIZE = 14;
        const int ACOL_SIZE = 3;
        int i = 0;
        int sheepAge;
        string sheepName;

        cout << left;
        cout << setw(NCOL_SIZE) << "NAME";
        cout << setw(NCOL_SIZE) << "AGE";
        cout << endl;
        cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
        cout << ' ';
        cout << setfill('-') << setw(ACOL_SIZE) << '-';
        cout << setfill(' ');
        cout << endl;
        while(i < sheepCount)
        {
            sheepArray[i].GetValues(sheepName, sheepAge);
            cout << setw(NCOL_SIZE) << sheepName;
            cout << setw(ACOL_SIZE) << sheepAge;
            cout << endl;

            i++;
        }
    }
    else
    {
        cout << "no sheep\n";
    }
}


```

```
************************************************
* PROGRAMMED BY : Faris Hijazi
* CLASS         : CS1A
* SECTION       : MW: 7:30P
* Assignment #13: Arrays and Linked Lists of Sheep - LL
************************************************

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: Fluffy
Sheep Age:  1

The sheep...
Sheep Name: Fluffy
Sheep Age:  1
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 ? 2

NAME          AGE
------------- ---
Fluffy        1

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: Maa
Sheep Age:  3

The sheep...
Sheep Name: Maa
Sheep Age:  3
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 ? 4

There are 2 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 ? 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

```

```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
 *****************************************************************************/

#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
};

```

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

```
/******************************************************************************
 * 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;
}
//------------------------------------------------------------------------------
```

```

/*******************************************************************************
 * METHOD GetName
 *------------------------------------------------------------------------------
 * Returns name
 * RETURN: name
 ******************************************************************************/
string Sheep::GetName() const
{
    return name;
}

//Farm methods

/*******************************************************************************
 * Farm CONSTRUCTOR
 *------------------------------------------------------------------------------
 * RETURN: NA
 ******************************************************************************/
Farm ::Farm()
{
    head = NULL;
    sheepCount = 0;
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * Farm DECONSTRUCTOR
 *------------------------------------------------------------------------------
 * RETURN: NA
 ******************************************************************************/
Farm ::~Farm()
{
    SheepNode *sheepPtr;

    sheepCount = 0;
    while(head != NULL)
    {
        sheepPtr = head;
        head = sheepPtr->next;
        delete sheepPtr;
    }
    sheepPtr = NULL;
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD AddSheep
 *------------------------------------------------------------------------------
 * adds a sheep to the linked list
 * RETURN: NA
 ******************************************************************************/
void Farm::AddSheep(Sheep newSheep)
{
    SheepNode *sheepPtr;
    SheepNode *tail;

    sheepCount++;
    sheepPtr = new SheepNode;
    if(head!=NULL)
    {
        tail = head;
        while(tail->next != NULL)
        {
```

```cpp
            tail = tail->next;
        }
        sheepPtr->next = NULL;
        tail->next = sheepPtr;
    }
    else
    {
        sheepPtr->next = head;
        head = sheepPtr;
    }
    sheepPtr->currentSheep = newSheep;
    sheepPtr = NULL;
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD ClearList
 *------------------------------------------------------------------------------
 * clears the list of all sheep
 * RETURN: NA
 ******************************************************************************/
void Farm::ClearList()
{
    SheepNode *sheepPtr;

    sheepCount = 0;
    while(head != NULL)
    {
        sheepPtr = head;
        head = sheepPtr->next;
        delete sheepPtr;
    }
    sheepPtr = NULL;
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD FindSheep
 *------------------------------------------------------------------------------
 * Finds sheep with specified name
 * RETURN: sheep to search
 ******************************************************************************/
Sheep Farm::FindSheep(string sheepName) const
{
    int i = 0;
    bool found = false;
    SheepNode *sheepPtr;
    if(sheepCount > 0)
    {
        sheepPtr = head;
        while(sheepPtr!=NULL && !found)
        {
            if (sheepPtr->currentSheep.GetName() == sheepName)
            {
                found = true;
            }
            else
            {
                sheepPtr = sheepPtr->next;
            }
        }
    }
    return(sheepPtr->currentSheep);
}
```

```cpp
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD GetFirstSheep
 *------------------------------------------------------------------------------
 * returns the first sheep
 * RETURN: first sheep
 ******************************************************************************/
Sheep Farm::GetFirstSheep() const
{
    return(head->currentSheep);
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD TotalSheep
 *------------------------------------------------------------------------------
 * returns total sheep
 * RETURN: sheepCount
 ******************************************************************************/
int Farm::TotalSheep() const
{
    return(sheepCount);
}
//------------------------------------------------------------------------------

/*******************************************************************************
 * METHOD DisplaySheepTable
 *------------------------------------------------------------------------------
 * displays all sheep
 * RETURN: NA
 ******************************************************************************/
void Farm::DisplaySheepTable() const
{
    if(sheepCount > 0)
    {
        const int NCOL_SIZE = 14;
        const int ACOL_SIZE = 3;
        int i = 0;
        int sheepAge;
        string sheepName;
        SheepNode *sheepPtr;

        cout << left;
        cout << setw(NCOL_SIZE) << "NAME";
        cout << setw(NCOL_SIZE) << "AGE";
        cout << endl;
        cout << setfill('-') << setw(NCOL_SIZE-1) << '-';
        cout << ' ';
        cout << setfill('-') << setw(ACOL_SIZE) << '-';
        cout << setfill(' ');
        cout << endl;

        sheepPtr = head;
        while(sheepPtr!=NULL)
        {
            sheepPtr->currentSheep.GetValues(sheepName, sheepAge);
            cout << setw(NCOL_SIZE) << sheepName;
            cout << setw(ACOL_SIZE) << sheepAge;
            cout << endl;

            sheepPtr = sheepPtr->next;
        }
    }
    else
    {
        cout << "no sheep\n";
    }
}
```