```
**************************************************
*  PROGRAMMED BY : ALI ESHGHI
*  CLASS         : CS1B
*  SECTION       : MW: 7:30p - 9:50p
*  LAB #14       : LAB 13 - Using inheritence classes
**************************************************

1 - Initialize Animals
0 - Exit
Enter Selection: 1

1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
0 - Exit
Enter Selection: 3

CHANGE AGE:
Which type of animal do you have in mind
1 - Sheep
2 - Pig
Select the animal type: 2

NAME          AGE    TAILTYPE
------------- ------ --------
Piggy          3      CURL_UP
Charlotte      10     CORKSCREW
Oink           2      STRAIGHT

Which Pig would you like to change:
Oink
Enter the new Age: 6
The Age for pig Oink has been changed to 6 !

1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
0 - Exit
Enter Selection: 1
Are you sure you want to reinitialize Sheeps (Y/N)?h

**** h is an invalid entry     ****
**** Please input Y or N ****
```

```
46
47 Are you sure you want to reinitialize Sheeps (Y/N)?n
48 Sheeps have not beenre-initialized!
49
50 1 - Re - Initializing Sheeps
51 2 - Re - Initializing Pigs
52 3 - Change Age
53 4 - Display
54 0 - Exit
55 Enter Selection: 2
56 Are you sure you want to reinitialize Pigs (Y/N)?r
57
58 **** r is an invalid entry     ****
59 **** Please input Y or N ****
60
61 Are you sure you want to reinitialize Pigs (Y/N)?n
62 Pigs have not beenre-initialized!
63
64 1 - Re - Initializing Sheeps
65 2 - Re - Initializing Pigs
66 3 - Change Age
67 4 - Display
68 0 - Exit
69 Enter Selection: 3
70
71 CHANGE AGE:
72 Which type of animal do you have in mind
73 1 - Sheep
74 2 - Pig
75 Select the animal type: 1
76
77 THE SHEEP:
78
79 NAME           AGE    WOOLTYPE COLOR
80 -------------- ------ -------- ----
81 Fluffy         5      MEDIUM   Black
82 Baa            7      FINE     White
83 Babe           8      LONG     Brown
84
85  Which Sheep would you like change:
86 Baa
87 Enter the new Age: 3
88 The Age for sheep Baa has been changed to 3 !
89
90 1 - Re - Initializing Sheeps
```

```
 2 - Re - Initializing Pigs
 3 - Change Age
 4 - Display
 0 - Exit
Enter Selection: 4

THE SHEEP:

NAME           AGE    WOOLTYPE COLOR
-------------- ------ -------- ----
Fluffy         5      MEDIUM   Black
Baa            3      FINE     White
Babe           8      LONG     Brown

THE PIG:

NAME          AGE    TAILTYPE
------------- ------ --------
Piggy          3      CURL_UP
Charlotte      10     CORKSCREW
Oink           6      STRAIGHT

1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
0 - Exit
Enter Selection: 5

**** The number 5 is an invalid entry     ****
**** Please input a number between 0 and 4 ****

1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
0 - Exit
Enter Selection: g

**** Please input a NUMBER between 0 and 4 ****

1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
```

```
0 - Exit
Enter Selection: 1
Are you sure you want to reinitialize Sheeps (Y/N)?y

Initializing the Sheeps...


1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
0 - Exit
Enter Selection: 2
Are you sure you want to reinitialize Pigs (Y/N)?y

Initializing the Pigs...


1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
0 - Exit
Enter Selection: 4

THE SHEEP:

NAME           AGE    WOOLTYPE COLOR
-------------- ------ -------- ----
Fluffy         5      MEDIUM   Black
Baa            7      FINE     White
Babe           8      LONG     Brown

THE PIG:

NAME          AGE    TAILTYPE
------------- ------ --------
Piggy          3      CURL_UP
Charlotte      10     CORKSCREW
Oink           2      STRAIGHT

1 - Re - Initializing Sheeps
2 - Re - Initializing Pigs
3 - Change Age
4 - Display
```

```
0 - Exit
Enter Selection: 0


```

```
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Julian Lasting
 * STUDENT ID : 1112261 & 1097778
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * LAB 14     : Farmer's Pete livestock(inheritence class)
 * DUE DATE   : 13 December 2019
 *****************************************************************************/

#ifndef MYHEADER_H_
#define MYHEADER_H_

#include<iostream>
#include<iomanip>
#include<string>
#include<fstream>
#include<limits>
#include<sstream>
#include "ClassHeader.h"
using namespace std;


enum Menu
{
    InitAnimal,
    InitSheep,
    InitPig,
    ChangeAge,
    Display,
    Exit
};


/*********************************************************
* CONSTANTS
* -------------------------------------------------------
* USED FOR CLASS HEADING - ALL WILL BE OUTPUT
* -------------------------------------------------------
* Type     : Program Type
* LAB_NUM  : Lab Number (specific to this lab)
* LAB_NAME : Title of the Lab
*********************************************************/

const string NAME = "LAB 13 - Using inheritence classes";
const char   TYPE    = 'L';
const int    NUM     = 14 ;
const string CLASS   = "CS1B";
const string SECTION = "MW: 7:30p - 9:50p";

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

```
 *               the function is void type
 ***************************************************************/
void PrintHeader();

/*********************************************************************************
 * FirstMenu
 *    This function gets the user choice for the first menu that has been run
 *
 *    RETURNS: integer
 *********************************************************************************/
Menu FirstMenu();

/*********************************************************************************
 * Menu
 *      This function will outputs the menu and prompts the user
 *      to choose an option from the menu
 *
 *      RETURN - integer
 *               the function is int type
 *********************************************************************************/
int MainMenu();

/*********************************************************************************
 * InitializeSheep
 *    This function gets the data from the input file for sheeps and then
 *    initializes the parrallel arrays with the information of sheeps
 *
 *
 *    RETURNS: nothing
 *             void type function
 *********************************************************************************/
void InitializeSheep(Animal &animal, Sheep &sheep);

/*********************************************************************************
 * InitializeSheep
 *    This function gets the data from the input file for pigs and then
 *    initializes the parrallel arrays with the information of pigs
 *
 *
 *    RETURNS: nothing
 *             void type function
 *********************************************************************************/
void InitializePig(Animal &animal, Pig &pig);

#endif /* MYHEADER_H_ */

```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Julian Lasting
 * STUDENT ID : 1112261 & 1097778
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * LAB 14     : Farmer's Pete livestock(inheritence class)
 * DUE DATE   : 13 December 2019
 *****************************************************************************/

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

/*****************************************************************************
* Lab 12
* ---------------------------------------------------------------------------
* This program uses the class and initializes the objects of the class using
* methods. then based on the user's choice the objects' values can be changed
* or get reinitialized to the first values.
* ---------------------------------------------------------------------------
* INPUT  : firstMenuOption -> The first choice that initializes the objects
*                             or exits the program
*          menuChoice      -> choice of the user to change the age or value,
*                             reinitialize, or display the data
*          newAge          -> change the value of the object age of the class
*          newValue        -> change the value of the object value of the class
* ---------------------------------------------------------------------------
* PROCESS: Initializing objects
*          Getting the first choice for the first menu
*          Getting the choice for the main menu
*          changing the age
*          changing the value
* ---------------------------------------------------------------------------
* OUTPUT : the values of the object variables of the class
*****************************************************************************/
int main()
{
    /*************************************************************************
    * CONSTANTS
    * ---------------------------------------------------------------------
    * OUTPUT - USED FOR CLASS HEADING
    * ---------------------------------------------------------------------
    * PROGRAMMER  : Programmer's Name
    * CLASS       : Student's Course
    * SECTION     : Class Days and Time
    * LAB_NUM     : Lab Number (specific to this lab)
    * LAB_NAME    : Title of the Assignment
    *************************************************************************/
    const string PROGRAMMER = "Ali Eshghi and Amirarsalan Valipour";
    const string CLASS      = "CS1B";
    const string SECTION= "MW: 7:30p - 9:50p";
    const int    LAB_NUM= 12;
    const string LAB_NAME   = "Intro to OOP";

    /*************
     * VARIABLES *
```

```
 56      *************/
 57
 58     Menu        startOption;  //IN - choice of the first menu
 59     int         menuOption;   //IN - choice of the main menu
 60     int         animalOption; //IN - choice of which animal
 61
 62
 63     bool    checkInp;     //PROCESS - input checking
 64
 65     char    initSure;
 66
 67     string      sheepName;
 68     string      pigName;
 69
 70     Menu    menuChoice;
 71
 72     Animal      animal;
 73     Sheep       sheep;
 74     Pig         pig;
 75
 76
 77
 78     /*************
 79      *  PROCESS  *
 80      *************/
 81     //this function will print header to the screen
 82     PrintHeader();
 83
 84
 85
 86
 87     //This function gets the user choice for the first menu to initialize or
 88     //exit the program
 89     startOption = FirstMenu();
 90
 91     switch(startOption)
 92     {
 93     case InitAnimal: menuChoice = InitAnimal;
 94                      InitializeSheep(animal, sheep);
 95                      InitializePig(animal, pig);
 96
 97     break;
 98     case Exit: menuChoice = Exit;
 99     break;
100     }
101
102     //while loop for the main menu untill the choice 0 is entered
103     while(menuChoice != Exit)
104     {
105         //this function will get the user's choice for the main menu
106         menuOption = MainMenu();
107
108         switch(menuOption)
109         {
110         case 1: menuChoice = InitSheep;
```

```
111         break;
112
113         case 2: menuChoice = InitPig;
114         break;
115
116         case 3: menuChoice = ChangeAge;
117         break;
118
119         case 4: menuChoice = Display;
120         break;
121
122         default: menuChoice = Exit;
123         }
124
125         //if statement for the first option
126         if(menuChoice == InitSheep)
127         {
128             checkInp = false;
129
130             do
131             {
132                 //INPUT - asks user if they are sure for reinitialization
133
134
135                 cout << "Are you sure you want to reinitialize Sheeps (Y/N)?";
136                 cin.get(initSure);
137
138
139                 //CHECKS FOR THE CHAR INPUT
140
141                 if (toupper(initSure) != 'Y' && toupper(initSure) != 'N')
142                 {
143                     cin.clear();
144                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
145
146                     cout << endl;
147                     cout << "**** "<< initSure
148                          << " is an invalid entry     ****" << endl;
149                     cout << "**** Please input Y or N ****";
150                     cout << endl << endl;
151
152                     checkInp = false;
153
154
155                 }
156
157                 else
158                 {
159
160                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
161                     checkInp = true;
162
163                 }
164
165             }while(!checkInp);
```

```
166
167             //if statement for reinitializing the classes to the first values
168             if(toupper(initSure) == 'Y')
169             {
170                 cout << "\nInitializing the Sheeps..." << endl
171                      << endl;
172
173
174
175                 //the function uses the methods to initialize the animal array
176                 InitializeSheep(animal, sheep);
177
178
179
180             }
181
182             else if(toupper(initSure) == 'N')
183             {
184                 cout << "Sheeps have not beenre-initialized!" << endl <<endl;
185             }
186         }
187
188
189         //if statement for the second option of the menu
190         else if(menuChoice == InitPig)
191         {
192             checkInp = false;
193
194             do
195             {
196                 //INPUT - asks user if they are sure for reinitialization
197
198
199                 cout << "Are you sure you want to reinitialize Pigs (Y/N)?";
200                 cin.get(initSure);
201
202
203                 //CHECKS FOR THE CHAR INPUT
204
205                 if (toupper(initSure) != 'Y' && toupper(initSure) != 'N')
206                 {
207                     cin.clear();
208                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
209
210                     cout << endl;
211                     cout << "**** "<< initSure
212                          << " is an invalid entry     ****" << endl;
213                     cout << "**** Please input Y or N ****";
214                     cout << endl << endl;
215
216                     checkInp = false;
217
218
219                 }
220
```

```
221                 else
222                 {
223 
224                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
225                     checkInp = true;
226 
227                 }
228 
229             }while(!checkInp);
230 
231             //if statement for reinitializing the classes to the first values
232             if(toupper(initSure) == 'Y')
233             {
234                 cout << "\nInitializing the Pigs..." << endl
235                      << endl;
236 
237                 //the function uses the methods to initialize the animal array
238                 InitializePig(animal, pig);
239 
240 
241 
242             }
243 
244             else if(toupper(initSure) == 'N')
245             {
246                 cout << "Pigs have not beenre-initialized!" << endl <<endl;
247             }
248 
249 
250 
251         }
252 
253 
254         //if statement for the third option of the main menu
255         else if(menuChoice == ChangeAge)
256         {
257             checkInp = false;
258 
259             cout << "\nCHANGE AGE:" << endl;
260 
261             //do while loop for user input for which animal type the user wants
262             //to change
263             do
264             {
265                 //INPUT
266 
267                 cout << "Which type of animal do you have in mind" << endl;
268 
269                 cout << "1 - Sheep" << endl;
270                 cout << "2 - Pig"    << endl;
271 
272                 cout << "Select the animal type: ";
273 
274 
275                 //CHECKS FOR THE CHAR INPUT
```

```
276
277                 if (!(cin >> animalOption))
278                 {
279                     cin.clear();
280                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
281
282                     cout << endl;
283                     cout << "**** Please input a NUMBER between 1 and 2 ****";
284                     cout << endl << endl;
285
286                     checkInp = false;
287
288                 }
289
290                 //CHECKS FOR THE RANGE ERROR
291
292                 else if (animalOption >= 3 || animalOption <= 0 )
293                 {
294
295                     cout << endl;
296                     cout << "**** The number "              << animalOption
297                     << " is an invalid entry     ****" << endl;
298                     cout << "**** Please input a number between 1 and 2 ****";
299                     cout << endl << endl;
300
301                     checkInp = false;
302
303                 }
304
305                 //PASS
306
307                 else
308                 {
309
310                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
311                     checkInp = true;
312
313                 }
314
315             }while(!checkInp);
316
317
318             //if statement to change the age for sheeps
319             if(animalOption == 1)
320             {
321
322                 //Display Sheep
323                 cout << endl;
324                 sheep.DisplayHeaderSheep();
325                 sheep.DisplaySheep();
326
327                 cout << "\n Which Sheep would you like change: " << endl;
328                 getline(cin,sheepName);
329
330                 sheep.ChangeSheepAge(sheepName);
```

```
331
332
333
334             }
335
336
337             //if statement to change the age for the pigs
338             else if(animalOption == 2)
339             {
340
341                 //Display Pig
342                 cout << endl;
343                 pig.DisplayHeaderPig();
344                 pig.DisplayPig();
345
346                 cout << "\nWhich Pig would you like to changeP: " << endl;
347                 getline(cin,pigName);
348
349                 pig.ChangePigAge(pigName);
350
351             }
352
353
354
355
356
357
358
359             //if statement if the user wants to change value for Babe
360             else if(menuChoice == Display)
361             {
362
363                 //these methods outputs the objects of the classes
364                 cout << endl;
365                 sheep.DisplayHeaderSheep();
366                 sheep.DisplaySheep();
367
368                 cout << endl;
369                 pig.DisplayHeaderPig();
370                 pig.DisplayPig();
371
372
373                 cout << endl << endl;
374             }
375         }
376
377
378         //Display
379
380     }
381
382
383     return 0;
384 }
385
```

386

```
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Julian Lasting
 * STUDENT ID : 1112261 & 1097778
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * LAB 14     : Farmer's Pete livestock(inheritence class)
 * DUE DATE   : 13 December 2019
 *****************************************************************************/

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

void InitializeSheep(Animal &animal, Sheep &sheep)
{
    string      sheepName;
    string      sheepAge;
    string      recordWool;
    string      recordColor;
    WoolType    woolType;

    int     index;

    ifstream    inFileSheep;


    inFileSheep.open("SheepFile.txt");

    sheep.~Sheep();

    index = 0;

    while(inFileSheep && index < AR_SIZE)
    {
        getline(inFileSheep,sheepName);

        getline(inFileSheep,sheepAge);

        //cin.ignore(1000,'\n');

        getline(inFileSheep, recordWool);

        getline(inFileSheep,recordColor);


        //cin.ignore(10000,'\n');

        if(recordWool == "LONG")
        {
            woolType = LONG;
        }

        else if(recordWool == "MEDIUM")
        {
            woolType = MEDIUM;
        }
```

```
56
57          else if(recordWool == "FINE")
58          {
59              woolType = FINE;
60          }
61
62          else if(recordWool == "CARPET")
63          {
64              woolType = CARPET;
65          }
66
67
68
69          sheep.GetName(sheepName);
70          sheep.SetSheepName(sheepName);
71
72          sheep.GetAge(sheepAge);
73          sheep.SetSheepAge(sheepAge);
74
75          sheep.SetWool(woolType);
76
77          sheep.SetWoolColor(recordColor);
78
79          sheep.DisplaySheep();
80
81
82          index++;
83      }
84
85      inFileSheep.close();
86  }
87
88
89
90
91  void InitializePig(Animal &animal, Pig &pig)
92  {
93      string      pigName;
94      string      pigAge;
95      string      recordTail;
96      TailType    tailType;
97
98      int     index;
99
100     ifstream    inFilePig;
101
102
103     inFilePig.open("PigFile.txt");
104
105
106
107     index = 0;
108
109
110     while(inFilePig && index < AR_SIZE)
```

```
111     {
112         getline(inFilePig,pigName);
113         getline(inFilePig,pigAge);
114         getline(inFilePig,recordTail);
115 
116         if(recordTail == "STRAIGHT")
117         {
118             tailType = STRAIGHT;
119         }
120 
121         else if(recordTail == "CORKSCREW")
122         {
123             tailType = CORKSCREW;
124         }
125 
126         else if(recordTail == "CURL_UP")
127         {
128             tailType = CURL_UP;
129         }
130 
131         else if(recordTail == "CURL_RIGHT")
132         {
133             tailType = CURL_RIGHT;
134         }
135 
136         else if(recordTail == "CURL_LEFT")
137         {
138             tailType = CURL_LEFT;
139         }
140 
141 
142 
143         //cin.ignore(10000,'\n');
144 
145         pig.GetName(pigName);
146         pig.SetPigName(pigName);
147         pig.GetAge(pigAge);
148         pig.SetPigAge(pigAge);
149         pig.SetTail(tailType);
150         pig.DisplayPig();
151 
152         index++;
153 
154     }
155 
156 
157     inFilePig.close();
158 }
159 
160 
161 /*****************************************************************************
162  * FirstMenu
163  *    This function gets the user choice for the first menu that has been run
164  *
165  *    RETURNS: integer
```

```
166 ********************************************************************************/
167
168 Menu FirstMenu()
169 {
170
171     /*************
172      * VARIABLES *
173      *************/
174
175     int startOption;
176     boolcheckInp;
177     Menuoption;
178
179     /***************
180      * INITIALIZE *
181      ***************/
182
183     checkInp = false;
184
185     //do while loop for error checking
186     do
187     {
188         //INPUT
189
190         cout << "1 - Initialize Animals " << endl;
191         cout << "0 - Exit"                << endl;
192         cout << "Enter Selection: ";
193
194
195         //CHECKS FOR THE CHAR INPUT
196
197         if (!(cin >> startOption))
198         {
199             cin.clear();
200             cin.ignore(numeric_limits<streamsize>::max(), '\n');
201
202             cout << endl;
203             cout << "**** Please input a NUMBER between 0 or 1 ****";
204             cout << endl << endl;
205
206             checkInp = false;
207
208         }
209
210         //CHECKS FOR THE RANGE ERROR
211
212         else if (startOption > 1 || startOption < 0 )
213         {
214
215             cout << endl;
216             cout << "**** The number "              << startOption
217             << " is an invalid entry     ****" << endl;
218             cout << "**** Please input a number between 0 or 1 ****";
219             cout << endl;
220
```

```
221             checkInp = false;
222 
223         }
224 
225         //PASS
226 
227         else
228         {
229 
230             cin.ignore(numeric_limits<streamsize>::max(), '\n');
231             checkInp = true;
232 
233         }
234 
235     }while(!checkInp);
236 
237     if(startOption == 1)
238     {
239         option = InitAnimal;
240     }
241 
242     else
243     {
244         option = Exit;
245     }
246 
247     //returns an integer to the main
248     return option;
249 }
250 
251 
252 
253 /*****************************************************************************
254  * Menu
255  * This function outputs the main menu and gets the user's choice for the menu
256  * options
257  *
258  * RETURNS: ineger
259  *****************************************************************************/
260 
261 int MainMenu()
262 {
263     /*************
264      * VARIABLES *
265      *************/
266 
267     int     menuOption; //IN - user input for menu
268     boolcheckInp;   //PROCESS - input check
269 
270     /***************
271      * INITIALIZE *
272      ***************/
273 
274     checkInp = false;
275 
```

```
276
277     //do while loop for error checking
278     do
279     {
280         //INPUT
281
282         cout << "1 - Re - Initializing Sheeps " << endl;
283         cout << "2 - Re - Initializing Pigs"    << endl;
284         cout << "3 - Change Age"            << endl;
285         cout << "4 - Display"                   << endl;
286         cout << "0 - Exit"                      << endl;
287
288         cout << "Enter Selection: ";
289
290
291         //CHECKS FOR THE CHAR INPUT
292
293         if (!(cin >> menuOption))
294         {
295             cin.clear();
296             cin.ignore(numeric_limits<streamsize>::max(), '\n');
297
298             cout << endl;
299             cout << "**** Please input a NUMBER between 0 and 4 ****";
300             cout << endl << endl;
301
302             checkInp = false;
303
304         }
305
306         //CHECKS FOR THE RANGE ERROR
307
308         else if (menuOption > 4 || menuOption < 0 )
309         {
310
311             cout << endl;
312             cout << "**** The number "              << menuOption
313             << " is an invalid entry     ****" << endl;
314             cout << "**** Please input a number between 0 and 4 ****";
315             cout << endl << endl;
316
317             checkInp = false;
318
319         }
320
321         //PASS
322
323         else
324         {
325
326             cin.ignore(numeric_limits<streamsize>::max(), '\n');
327             checkInp = true;
328
329         }
330
```

```
331     }while(!checkInp);
332
333     //returns an integer to the main
334     return menuOption;
335 }
336
337 /*************************************************************
338  * PrintHeaderFile
339  * This function will output the class heading to the screen.
340  *
341  * return type - nothing
342  *               the function is void type
343  *************************************************************/
344
345 void PrintHeader()
346 {
347     cout << left;
348     cout << "*****************************************************\n"       ;
349     cout << "*  PROGRAMMED BY : ALI ESHGHI"                                ;
350     cout << "\n*  "     << setw(14) << "CLASS"     << ": " << CLASS       ;
351     cout << "\n*  "     << setw(14) << "SECTION"   << ": " << SECTION     ;
352     cout << "\n*  LAB #"<< setw(9)  << NUM  << ": " << NAME              ;
353     cout << "\n*****************************************************\n\n"   ;
354     cout << right;
355 }
356
357
358
359
```

```
 1 /*****************************************************************************
 2  * PROGRAMMER : Ali Eshghi & Julian Lasting
 3  * STUDENT ID : 1112261 & 1097778
 4  * CLASS      : CS1B
 5  * SECTION    : MW 7:30pm
 6  * LAB 14     : Farmer's Pete livestock(inheritence class)
 7  * DUE DATE   : 13 December 2019
 8  *****************************************************************************/
 9
10 #ifndef CLASSHEADER_H_
11 #define CLASSHEADER_H_
12
13 #include<iostream>
14 #include<iomanip>
15 #include<string>
16 #include<fstream>
17 #include<limits>
18 #include<sstream>
19 using namespace std;
20
21 const int AR_SIZE = 3;
22
23
24
25 enum WoolType
26 {
27     LONG,
28     MEDIUM,
29     FINE,
30     CARPET
31 };
32
33
34 enum TailType
35 {
36     STRAIGHT,
37     CORKSCREW,
38     CURL_UP,
39     CURL_RIGHT,
40     CURL_LEFT
41 };
42
43
44
45
46
47 class Animal
48 {
49     //public part of the class that is available for outside of the class
50     public:
51         //constructor
52         Animal();
53
54         //destructor
55         ~Animal();
```

```
56
57          //Method for adding a new animal's name.
58          void GetName(string name);
59
60          //Method for adding a new animal's age.
61          void GetAge(string age);
62
63          //method for changing the age
64          void changeAge();
65
66          //method for changing the name
67          void changeName();
68
69          //method(overload)for changing the name and age
70          void changeAgeName();
71
72          //method that returns the animal's name
73          string setName() const;
74
75          //method that returns the animal's age
76          int setAge() const;
77
78          //method  for outputting the objects
79          void Display() const;
80
81          //method to get the list size of the animals in the list
82          int GetAnimalCount() const;
83
84
85
86      //private part only available for this class(in this case the attributes
87      //                                          can be used in derived classes)
88      private:
89          string  nameAr[AR_SIZE];
90          string  ageAr[AR_SIZE];
91          int     animalCount;
92  };
93
94
95  class Sheep: public Animal  //derived class
96  {
97      //public part of the class that is available for outside of the class
98      public:
99          //constructor
100         Sheep();
101
102         //decosntructor
103         ~Sheep();
104
105         //method to get the name of the sheep from the file and put in array
106         void SetSheepName(string name);
107
108         //method to get the age of the sheep from the file and put in array
109         void SetSheepAge(string age);
110
```

```
111         //method to set the wool type
112         void SetWool(WoolType wool);
113 
114         //method to set the wool color
115         void SetWoolColor(string color);
116 
117         //method to change the age for the sheeps
118         void ChangeSheepAge(string name);
119 
120         //method to show the header for the diplay
121         void DisplayHeaderSheep() const;
122 
123         //method to display the sheep
124         void DisplaySheep();
125 
126         //method to get the wool type
127         WoolType GetWool() const;
128 
129         //method that returns the size of the list of the sheeps
130         int GetSheepCount() const;
131 
132 
133     //private part only available for this class(in this case the attributes
134     //                                           can be used in derived classes)
135     private:
136         string       nameAr[AR_SIZE] = {" "};
137         string       ageAr[AR_SIZE] = {" "};
138         WoolTypewoolAr[AR_SIZE] = {LONG};
139         string       colorAr[AR_SIZE] = {" "};
140         int          sheepCount;
141 
142 
143 };
144 
145 
146 
147 
148 class Pig: public Animal
149 {
150     //public part of the class that is available for outside of the class
151     public:
152         //constructor
153         Pig();
154 
155         //decosntructor
156         ~Pig();
157 
158         //method to get the name of the pig from the file and put it in array
159         void SetPigName(string name);
160 
161         //method to get the age of the pig from the file and put it in array
162         void SetPigAge(string age);
163 
164         //method to set the tail Type
165         void SetTail(TailType tail);
```

```
166
167         //method to show header for the display
168         void DisplayHeaderPig() const;
169
170         //method to display pig
171         void DisplayPig() const;
172
173         //method to get the tail type
174         TailType GetTail() const;
175
176         //method to change the age for pig
177         void ChangePigAge(string name);
178
179         //method that returns the size of the list of the sheeps
180         int GetPigCount() const;
181
182         //method for finding the Pig in the list
183         void FindPig(string) const;
184
185
186     //private part only available for the class
187     private:
188         string      nameAr[AR_SIZE] = {"p"};
189         string  ageAr[AR_SIZE] = {"0"};
190         TailType    tailAr[AR_SIZE] = {STRAIGHT};
191         int     pigCount;
192
193
194 };
195
196
197
198 #endif /* CLASSHEADER_H_ */
199
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi & Julian Lasting
 * STUDENT ID : 1112261 & 1097778
 * CLASS      : CS1B
 * SECTION    : MW 7:30pm
 * LAB 14     : Farmer's Pete livestock(inheritence class)
 * DUE DATE   : 13 December 2019
 *****************************************************************************/

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


/*****************************************************************************
 * METHODS FOR CLASS ANIMAL
 *****************************************************************************/

Animal::Animal()    /*** CONSTRUCTOR ***/
{
    animalCount = 0;
}

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

//method to retrieve the name from the file
void Animal::GetName(string name)           /***  MUTATORS ***/
{
    //verify wether the array is not full
    if(animalCount < AR_SIZE)
    {
        //set the data in the array
        nameAr[animalCount] = name;

        //update the animal counter with one more class
        animalCount++;
    }

    else
    {
        cout << "Could not add animal - array is full" << endl;
    }
}

//method to retrieve the age of the animal from the file
void Animal::GetAge(string age)         /***  MUTATORS ***/
{
    //verify wether the array is not full
    if(animalCount < AR_SIZE)
    {
        //set the data in the array
        ageAr[animalCount] = age;
    }

}

```

```
56 void Animal::Display() const
57 {
58
59 }
60
61 /*****************************************************************************
62  * METHODS FOR CLASS Sheep
63  *****************************************************************************/
64
65 Sheep::Sheep()
66 {
67     sheepCount = 0;
68 }
69
70 Sheep::~Sheep() {}
71
72 //method to retrieve the name from the file
73 void Sheep::SetSheepName(string name)           /***  MUTATORS ***/
74 {
75     //verify wether the array is not full
76     if(sheepCount < AR_SIZE)
77     {
78         //set the data in the array
79         nameAr[sheepCount] = name;
80
81         //update the animal counter with one more class
82         //sheepCount++;
83     }
84
85     else
86     {
87         cout << "Could not add animal - array is full" << endl;
88     }
89 }
90
91 //method to retrieve the age of the animal from the file
92 void Sheep::SetSheepAge(string age)          /***  MUTATORS ***/
93 {
94     //verify wether the array is not full
95     if(sheepCount < AR_SIZE)
96     {
97         //set the data in the array
98         ageAr[sheepCount] = age;
99     }
100
101 }
102
103
104
105 void Sheep::SetWool(WoolType wool)
106 {
107
108     if(sheepCount < AR_SIZE)
109     {
110
```

```
111         if(wool == LONG)
112         {
113             woolAr[sheepCount] = LONG;
114         }
115 
116         else if(wool == MEDIUM)
117         {
118             woolAr[sheepCount] = MEDIUM;
119         }
120 
121         else if(wool == FINE)
122         {
123             woolAr[sheepCount] = FINE;
124         }
125 
126         else if(wool == CARPET)
127         {
128             woolAr[sheepCount] = CARPET;
129         }
130     }
131 
132     else
133     {
134         cout << "Could not add animal - array is full" << endl;
135     }
136 
137 }
138 
139 void Sheep::SetWoolColor(string color)
140 {
141     if(sheepCount < AR_SIZE)
142     {
143         colorAr[sheepCount] = color;
144 
145         sheepCount++;
146     }
147 
148 }
149 
150 void Sheep::DisplayHeaderSheep() const
151 {
152     cout << left;
153     cout << "THE SHEEP:" << endl << endl;
154     cout << setw(15) << "NAME" << setw(7)
155          << "AGE" << setw(9) << "WOOLTYPE" << setw(5) << "COLOR"
156          <<endl;
157 
158     cout << "--------------- " << "------ " << "-------- "
159          << "----" << endl;
160 
161 }
162 
163 void Sheep::DisplaySheep()
164 {
165     int i;
```

```
166
167     for(i = 0; i < AR_SIZE; i++)
168     {
169         cout << left;
170         cout << setw(15) << nameAr[i] << setw(7) << ageAr[i] << setw(9);
171
172         if(woolAr[i] == LONG)
173         {
174             cout << "LONG";
175         }
176
177         else if(woolAr[i] == MEDIUM)
178         {
179             cout << "MEDIUM";
180         }
181
182         else if(woolAr[i] == FINE)
183         {
184             cout << "FINE";
185         }
186
187         else if(woolAr[i] == CARPET)
188         {
189             cout << "CARPET";
190         }
191
192         cout << setw(5) << colorAr[i] << endl;
193     }
194 }
195
196 void Sheep::ChangeSheepAge(string name)
197 {
198     int i;
199     int newAge;
200     bool    check;
201     bool    checkInp;
202
203     check = false;
204     i = 0;
205
206     while(!check && i < AR_SIZE)
207     {
208         if(nameAr[i] == name)
209         {
210             check = true;
211         }
212
213         else
214         {
215             i++;
216         }
217     }
218
219     if(check == true)
220     {
```

```
221         do
222         {
223             //INPUT
224             cout << "Enter the new Age: ";
225 
226 
227             //CHECKS FOR THE CHAR INPUT
228 
229             if (!(cin >> newAge))
230             {
231                 cin.clear();
232                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
233 
234                 cout << endl;
235                 cout << "**** Please input a NUMBER between 0 and 10 ****";
236                 cout << endl << endl;
237 
238                 checkInp = false;
239 
240             }
241 
242             //CHECKS FOR THE RANGE ERROR
243 
244             else if (newAge >= 11 || newAge <= -1 )
245             {
246 
247                 cout << endl;
248                 cout << "**** The number "              << newAge
249                 << " is an invalid entry     ****" << endl;
250                 cout << "**** Please input a number between 0 and 10 ****";
251                 cout << endl << endl;
252 
253                 checkInp = false;
254 
255             }
256 
257             //PASS
258 
259             else
260             {
261 
262                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
263                 checkInp = true;
264 
265             }
266 
267         }while(!checkInp);
268 
269         ageAr[i] = newAge;
270         cout << "The Age for sheep " << nameAr[i] << " has been changed to "
271              << newAge << " !" << endl << endl;
272     }
273 
274     else
275     {
```

```
276         cout << "Could not find the sheep " << name << " to change the age"
277              << endl;
278     }
279 
280 }
281 
282 /*
283 WoolType Sheep::GetWool() const;
284 {
285     return wool;
286 }
287 */
288 
289 /*****************************************************************************
290  * METHODS FOR CLASS Pig
291  *****************************************************************************/
292 
293 Pig::Pig()
294 {
295     pigCount = 0;
296 }
297 
298 Pig::~Pig()
299 {
300     pigCount = 0;
301 }
302 
303 //method to retrieve the name from the file
304 void Pig::SetPigName(string name)           /***  MUTATORS ***/
305 {
306     //verify wether the array is not full
307     if(pigCount < AR_SIZE)
308     {
309         //set the data in the array
310         nameAr[pigCount] = name;
311 
312     }
313 
314     else
315     {
316         cout << "Could not add animal - array is full" << endl;
317     }
318 }
319 
320 //method to retrieve the age of the animal from the file
321 void Pig::SetPigAge(string age)         /***  MUTATORS ***/
322 {
323     //verify wether the array is not full
324     if(pigCount < AR_SIZE)
325     {
326         //set the data in the array
327         ageAr[pigCount] = age;
328     }
329 
330 }
```

```
331
332
333
334 void Pig::SetTail(TailType tail)
335 {
336     if(pigCount < AR_SIZE)
337     {
338         switch(tail)
339         {
340         case STRAIGHT:   tailAr[pigCount] = STRAIGHT;
341
342         break;
343
344         case CORKSCREW:  tailAr[pigCount] = CORKSCREW;
345
346         break;
347
348         case CURL_UP:    tailAr[pigCount] = CURL_UP;
349
350         break;
351
352         case CURL_RIGHT: tailAr[pigCount] = CURL_RIGHT;
353
354         break;
355
356         case CURL_LEFT: tailAr[pigCount] = CURL_LEFT;
357
358         break;
359         }
360
361         pigCount++;
362     }
363
364
365
366 }
367
368 void Pig::DisplayHeaderPig() const
369 {
370     cout << "THE PIG" << endl;
371
372     cout << setw(14) << "NAME" << setw(7)
373          << "AGE" << setw(9) << "TAILTYPE"
374          << endl;
375
376     cout << "-------------- " << "------ " << "-------- " << endl;
377
378 }
379
380 void Pig::DisplayPig() const
381 {
382     int i;
383
384     for(i = 0; i < AR_SIZE; i++)
385     {
```

```
386         cout << left;
387         cout << setw(15) << nameAr[i] << setw(7) << ageAr[i] << setw(9);
388 
389         if(tailAr[i] == STRAIGHT)
390         {
391             cout << "STRAIGHT";
392         }
393 
394         else if(tailAr[i] == CORKSCREW)
395         {
396             cout << "CORKSCREW";
397         }
398 
399         else if(tailAr[i] == CURL_UP)
400         {
401             cout << "CURL_UP";
402         }
403 
404         else if(tailAr[i] == CURL_RIGHT)
405         {
406             cout << "CURL_RIGHT";
407         }
408 
409         else if(tailAr[i] == CURL_LEFT)
410         {
411             cout << "CURL_LEFT";
412         }
413 
414         cout << endl;
415     }
416 }
417 
418 
419 
420 void Pig::ChangePigAge(string name)
421 {
422     int i;
423     int newAge;
424     bool check;
425     bool checkInp;
426 
427     check = false;
428     i = 0;
429 
430     while(!check && i < AR_SIZE)
431     {
432         if(nameAr[i] == name)
433         {
434             check = true;
435         }
436 
437         else
438         {
439             i++;
440         }
```

```
441     }
442
443     if(check == true)
444     {
445         do
446         {
447             //INPUT
448             cout << "Enter the new Age: ";
449
450
451             //CHECKS FOR THE CHAR INPUT
452
453             if (!(cin >> newAge))
454             {
455                 cin.clear();
456                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
457
458                 cout << endl;
459                 cout << "**** Please input a NUMBER between 0 and 10 ****";
460                 cout << endl << endl;
461
462                 checkInp = false;
463
464             }
465
466             //CHECKS FOR THE RANGE ERROR
467
468             else if (newAge >= 11 || newAge <= -1 )
469             {
470
471                 cout << endl;
472                 cout << "**** The number "              << newAge
473                 << " is an invalid entry     ****" << endl;
474                 cout << "**** Please input a number between 0 and 10 ****";
475                 cout << endl << endl;
476
477                 checkInp = false;
478
479             }
480
481             //PASS
482
483             else
484             {
485
486                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
487                 checkInp = true;
488
489             }
490
491         }while(!checkInp);
492
493         ageAr[i] = newAge;
494         cout << "The Age for pig " << nameAr[i] << " has been changed to "
495              << newAge << " !" << endl << endl;
```

```
496     }
497
498     else
499     {
500         cout << "Could not find the pig " << name << " to change the age"
501              << endl;
502     }
503
504 }
505
506
507 /*
508 TailType Pig::GetTail() const;
509 {
510     return tail;
511 }
512 */
513
```