```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
```