```
************************************************************
*   PROGRAMMED BY : Andrew Gharios
*   STUDENT ID    : 1449366
*   CLASS         : M-Th 5-7:20p
*   LAB #14       : Intro Inheritance, Overloading & Redefining OOP
************************************************************

1 - Initialize Animals
0 - Exit
Enter selection: 4

**** The number 4 is an invalid entry     *****
**** Please input a number between 0 and 1 *****

Enter selection: 1

1 - Re-Initialize Sheep
2 - Re-Initialize Pigs
3 - Change Age
4 - Display
0 - Exit

Enter selection: 4

THE SHEEP:

NAME           AGE WOOLTYPE COLOR
-------------- --- -------- -----
Fluffy          1  Fine     White
Maa             8  Carpet   Brown
La La           3  Long     Black

THE PIG(S):

NAME           AGE TAILTYPE
-------------- --- ---------
Babe            4  Corkscrew
Wilbur          7  Curl Up
KiKi            2  Straight

Enter selection: 3

Would you like to set the age of sheep or pigs? cats

**** cats is an invalid entry ****
**** Please input (sheep or pigs)   ****

Would you like to set the age of sheep or pigs? sheep

```

```
1 - Fluffy
2 - Maa
3 - La La

Select the animal you'd like to change: 2

NEW AGE: 11

**** The number 11 is an invalid entry     *****
**** Please input a number between 0 and 10 *****

NEW AGE: -1

**** The number -1 is an invalid entry     *****
**** Please input a number between 0 and 10 *****

NEW AGE: 5
Changing Maa's age to 5 ...

Enter selection: 4

THE SHEEP:

NAME           AGE WOOLTYPE COLOR
-------------- --- -------- -----
Fluffy          1  Fine     White
Maa             5  Carpet   Brown
La La           3  Long     Black

THE PIG(S):

NAME           AGE TAILTYPE
-------------- --- ---------
Babe            4  Corkscrew
Wilbur          7  Curl Up
KiKi            2  Straight

Enter selection: 3

Would you like to set the age of sheep or pigs? pigs

1 - Babe
2 - Wilbur
3 - KiKi

Select the animal you'd like to change: 3

NEW AGE: d

```

```
**** Please enter a NUMBER between 0 and 10 ****

NEW AGE: 7
Changing KiKi's age to 7 ...

Enter selection: 4

THE SHEEP:

NAME           AGE WOOLTYPE COLOR
-------------- --- -------- -----
Fluffy          1  Fine     White
Maa             5  Carpet   Brown
La La           3  Long     Black

THE PIG(S):

NAME           AGE TAILTYPE
-------------- --- ---------
Babe            4  Corkscrew
Wilbur          7  Curl Up
KiKi            7  Straight

Enter selection: 2

The Pig(s) have been reinitialized.

Enter selection: 4

THE SHEEP:

NAME           AGE WOOLTYPE COLOR
-------------- --- -------- -----
Fluffy          1  Fine     White
Maa             5  Carpet   Brown
La La           3  Long     Black

THE PIG(S):

NAME           AGE TAILTYPE
-------------- --- ---------
Babe            4  Corkscrew
Wilbur          7  Curl Up
KiKi            2  Straight

Enter selection: 1

The Sheep have been reinitialized.

```

```
Enter selection: 4

THE SHEEP:

NAME           AGE WOOLTYPE COLOR
-------------- --- -------- -----
Fluffy          1  Fine     White
Maa             8  Carpet   Brown
La La           3  Long     Black

THE PIG(S):

NAME           AGE TAILTYPE
-------------- --- ---------
Babe            4  Corkscrew
Wilbur          7  Curl Up
KiKi            2  Straight

Enter selection: 0
```

```
1  #ifndef HEADER_H_
2  #define HEADER_H_
3
4  #include <iostream> // cin, cout.
5  #include <string>   // string datatype variables.
6  #include <fstream>  // Fstream files.
7  #include <iomanip>  // fixed, setw, setprecision.
8  #include <ostream>  // Ostream data type.
9  #include <ctype.h>
10
11 using namespace std;
12
13 /*****************************************************************************
14 * CONSTANTS SETW SIZE
15 * ---------------------------------------------------------------------------
16 * NAME_SIZE
17 * AGE_SIZE
18 * WOOL_SIZE
19 *****************************************************************************/
20 const int NAME_SIZE = 15;
21 const int AGE_SIZE = 4;
22 const int WOOL_SIZE = 9;
23
24
25 /*****************************************************************************
26  * PrintHeaderFile
27  *   This function will output the header information
28  *
29                                                                              ↵
   *****************************************************************************/
30 void PrintHeaderFile(ostream& output,      // IN  - output datatype.
31     string asName,        // IN  - assignment name
32     int asNum,             // IN  - assignment number
33     string studentName,   // IN  - student's name
34     string classInfo,     // IN  - class that is being taken
35     char asType,          // IN  - assignment type
36     long long studentID); // IN  - student ID
37
38
39 #endif
40
41
```

```cpp
/*****************************************************************************
* AUTHOR     : Andrew Gharios
* STUDENT ID : 1449366
* LAB #14    : Inheritance, Overloading & Redefining
* CLASS      : CS1B
* SECTION    : M-TH: 5-7:20p
* DUE DATE   : 7/23/21
*****************************************************************************/
#include "Header.h"
#include "Animal.h"
#include "Sheep.h"
#include "Pig.h"

/*****************************************************************************
* Inheritance, Overloading, & Redefining
*----------------------------------------------------------------------------
* This program will allow the user to manipulate 3 sheep and 3 pigs, the
* program will use Inheritance, Overloading, & Redefining techniques allowing
* the user to initialize the pig, or the sheep, display all the animals with
* their specifications or change the age of an animal of user's choice.
*----------------------------------------------------------------------------
* INPUT:
* input        - Main menu input.
* inp          - Selection of which animal to modify.
* ageInp   - New age for animal.
* animalInp    - Type of animal user wants to change age for.
*****************************************************************************/
int main()
{
    /*************************************************************************
    * CONSTANTS
    *-------------------------------------------------------------------------
    * OUTPUT - USED FOR CLASS HEADING
    *-------------------------------------------------------------------------
    * PROGRAMMER : Programmer's Name
```

```
36      * CLASS   : Student's Course
37      * SECTION    : Class Days and Times
38      * LAB_NUM    : Lab Number (specific to this lab)
39      * LAB_NAME   : Title of the Lab
40      * -------------------------------------------------------------------------
41      * SetW Sizes
42      * -------------------------------------------------------------------------
43      * NAME_SIZE
44      * AGE_SIZE
45      * WOOL_SIZE
46      ************************************************************************** ➲
          /
47
48      const string AS_NAME = "Intro Inheritance, Overloading & Redefining OOP";
49      const int AS_NUM = 14;
50      const string STUDENT_NAME = "Andrew Gharios";
51      const string CLASS_INFO = "M-Th 5-7:20p";
52      const char AS_TYPE = 'L';
53      const long long STUDENT_ID = 1449366;
54
55      ifstream inFile; // CALC   - Input file variable.
56      Sheep fluffy;        // CALC & OUT - sheep #1
57      Sheep lala;          // CALC & OUT - sheep #2
58      Sheep maa;           // CALC & OUT - sheep #3
59      Pig   babe;          // CALC & OUT - Pig #1
60      Pig   wilbur;        // CALC & OUT - Pig #2
61      Pig   kiki;          // CALC & OUT - Pig #3
62      WoolType wool;       // CALC   - Sheep's wooltype.
63      TailType tail;       // CALC   - Pig's tailtype.
64      int    input;        // IN & CALC  - Menu selection.
65      bool   invalid;      // CALC   - Input validation.
66      int    inp;          // IN & CALC  - Animal selection.
67      string animalInp;    // IN & CALC  - If user wants to reinitialize.
68      string nameInp;      // IN & CALC  - Name input.
69      int    ageInp;       // IN & CALC  - New age for selected animal.
70      string woolCol;      // IN & CALC  - Wool color input.
71      string tailInp;      // IN & CALC  - Tail input.
72      string woolInp;      // IN & CALC  - Wool input.
73
74      invalid = false;
75
76
77      PrintHeaderFile(cout, AS_NAME, AS_NUM, STUDENT_NAME, CLASS_INFO,
78          AS_TYPE, STUDENT_ID);
79
80      cout << "\n1 - Initialize Animals\n";
81      cout << "0 - Exit";
82
83
```

```
 84      do
 85      {
 86          invalid = false;
 87          cout << "\nEnter selection: ";
 88          if (!(cin >> input))
 89          {
 90              cout << "\n**** Please enter a NUMBER between 0 and 1 ****\n";
 91              cin.clear();
 92              cin.ignore(numeric_limits<streamsize>::max(), '\n');
 93              invalid = true;
 94          }
 95          else if (input < 0 || input > 1)
 96          {
 97
 98              cout << "\n**** The number " << input << " is an invalid entry
                    *****\n";
 99              cout << "**** Please input a number between 0 and 1 *****\n";
100              invalid = true;
101          }
102
103
104      } while (invalid);
105
106      cin.ignore(numeric_limits<streamsize>::max(), '\n');
107
108      if (input == 1)
109      {
110          /*** SHEEP ***/
111          inFile.open("SheepInput.txt");
112          getline(inFile, nameInp);
113          inFile >> ageInp;
114          inFile.ignore(numeric_limits<streamsize>::max(), '\n');
115          fluffy.SetInitialValues(nameInp, ageInp);
116          getline(inFile, woolInp);
117          getline(inFile, woolCol);
118          if (woolInp == "LONG")
119          {
120              wool = LONG;
121          }
122          else if (woolInp == "MEDIUM")
123          {
124              wool = MEDIUM;
125          }
126          else if (woolInp == "FINE")
127          {
128              wool = FINE;
129          }
130          else if (woolInp == "CARPET")
131          {
```

```
132             wool = CARPET;
133         }
134         fluffy.SetWool(wool, woolCol);
135         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
136
137         getline(inFile, nameInp);
138         inFile >> ageInp;
139         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
140         maa.SetInitialValues(nameInp, ageInp);
141         getline(inFile, woolInp);
142         getline(inFile, woolCol);
143         if (woolInp == "LONG")
144         {
145             wool = LONG;
146         }
147         else if (woolInp == "MEDIUM")
148         {
149             wool = MEDIUM;
150         }
151         else if (woolInp == "FINE")
152         {
153             wool = FINE;
154         }
155         else if (woolInp == "CARPET")
156         {
157             wool = CARPET;
158         }
159         maa.SetWool(wool, woolCol);
160         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
161
162         getline(inFile, nameInp);
163         inFile >> ageInp;
164         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
165         lala.SetInitialValues(nameInp, ageInp);
166         getline(inFile, woolInp);
167         getline(inFile, woolCol);
168         if (woolInp == "LONG")
169         {
170             wool = LONG;
171         }
172         else if (woolInp == "MEDIUM")
173         {
174             wool = MEDIUM;
175         }
176         else if (woolInp == "FINE")
177         {
178             wool = FINE;
179         }
180         else if (woolInp == "CARPET")
```

```
181         {
182             wool = CARPET;
183         }
184         lala.SetWool(wool, woolCol);
185         inFile.close();
186 
187 
188         // *** PIG *** /
189         inFile.open("PigInput.txt");
190         getline(inFile, nameInp);
191         inFile >> ageInp;
192         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
193         babe.SetInitialValues(nameInp, ageInp);
194         getline(inFile, tailInp);
195         if (tailInp == "STRAIGHT")
196         {
197             tail = STRAIGHT;
198         }
199         else if (tailInp == "CORKSCREW")
200         {
201             tail = CORKSCREW;
202         }
203         else if (tailInp == "CURL_UP")
204         {
205             tail = CURL_UP;
206         }
207         else if (tailInp == "CURL_RIGHT")
208         {
209             tail = CURL_RIGHT;
210         }
211         else if (tailInp == "CURL_LEFT")
212         {
213             tail = CURL_LEFT;
214         }
215         babe.SetTail(tail);
216         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
217 
218         getline(inFile, nameInp);
219         inFile >> ageInp;
220         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
221         wilbur.SetInitialValues(nameInp, ageInp);
222         getline(inFile, tailInp);
223         if (tailInp == "STRAIGHT")
224         {
225             tail = STRAIGHT;
226         }
227         else if (tailInp == "CURL_UP")
228         {
229             tail = CURL_UP;
```

```
230         }
231         else if (tailInp == "CURL_RIGHT")
232         {
233             tail = CURL_RIGHT;
234         }
235         else if (tailInp == "CURL_LEFT")
236         {
237             tail = CURL_LEFT;
238         }
239         wilbur.SetTail(tail);
240         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
241
242         getline(inFile, nameInp);
243         inFile >> ageInp;
244         inFile.ignore(numeric_limits<streamsize>::max(), '\n');
245         kiki.SetInitialValues(nameInp, ageInp);
246         getline(inFile, tailInp);
247         if (tailInp == "STRAIGHT")
248         {
249             tail = STRAIGHT;
250         }
251         else if (tailInp == "CURL_UP")
252         {
253             tail = CURL_UP;
254         }
255         else if (tailInp == "CURL_RIGHT")
256         {
257             tail = CURL_RIGHT;
258         }
259         else if (tailInp == "CURL_LEFT")
260         {
261             tail = CURL_LEFT;
262         }
263         kiki.SetTail(tail);
264         inFile.close();
265
266         cout << "\n1 - Re-Initialize Sheep\n";
267         cout << "2 - Re-Initialize Pigs\n";
268         cout << "3 - Change Age\n";
269         cout << "4 - Display\n";
270         cout << "0 - Exit\n";
271
272         do
273         {
274             do
275             {
276                 invalid = false;
277                 cout << "\nEnter selection: ";
278                 if (!(cin >> input))
```

```
279                 {
280                     cout << "\n**** Please enter a NUMBER between 0 and 4 ****
                    \n";
281                     cin.clear();
282                     cin.ignore(numeric_limits<streamsize>::max(), '\n');
283                     invalid = true;
284                 }
285                 else if (input < 0 || input > 4)
286                 {
287
288                     cout << "\n**** The number " << input << " is an invalid
                    entry     *****\n";
289                     cout << "**** Please input a number between 0 and 4 *****
                    \n";
290                     invalid = true;
291                 }
292
293
294             } while (invalid);
295             cin.ignore(numeric_limits<streamsize>::max(), '\n');
296
297             switch (input)
298             {
299                 //RE-INITIALIZE SHEEP
300             case 1:
301                 //sheep
302                 cout << "\nThe Sheep have been reinitialized.\n";
303                 inFile.open("SheepInput.txt");
304                 getline(inFile, nameInp);
305                 inFile >> ageInp;
306                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
307                 fluffy.SetInitialValues(nameInp, ageInp);
308                 getline(inFile, woolInp);
309                 getline(inFile, woolCol);
310                 if (woolInp == "LONG")
311                 {
312                     wool = LONG;
313                 }
314                 else if (woolInp == "MEDIUM")
315                 {
316                     wool = MEDIUM;
317                 }
318                 else if (woolInp == "FINE")
319                 {
320                     wool = FINE;
321                 }
322                 else if (woolInp == "CARPET")
323                 {
324                     wool = CARPET;
```

```
325                 }
326                 fluffy.SetWool(wool, woolCol);
327                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
328
329                 getline(inFile, nameInp);
330                 inFile >> ageInp;
331                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
332                 maa.SetInitialValues(nameInp, ageInp);
333                 getline(inFile, woolInp);
334                 getline(inFile, woolCol);
335                 if (woolInp == "LONG")
336                 {
337                     wool = LONG;
338                 }
339                 else if (woolInp == "MEDIUM")
340                 {
341                     wool = MEDIUM;
342                 }
343                 else if (woolInp == "FINE")
344                 {
345                     wool = FINE;
346                 }
347                 else if (woolInp == "CARPET")
348                 {
349                     wool = CARPET;
350                 }
351                 maa.SetWool(wool, woolCol);
352                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
353
354                 getline(inFile, nameInp);
355                 inFile >> ageInp;
356                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
357                 lala.SetInitialValues(nameInp, ageInp);
358                 getline(inFile, woolInp);
359                 getline(inFile, woolCol);
360                 if (woolInp == "LONG")
361                 {
362                     wool = LONG;
363                 }
364                 else if (woolInp == "MEDIUM")
365                 {
366                     wool = MEDIUM;
367                 }
368                 else if (woolInp == "FINE")
369                 {
370                     wool = FINE;
371                 }
372                 else if (woolInp == "CARPET")
373                 {
```

```
374                     wool = CARPET;
375                 }
376                 lala.SetWool(wool, woolCol);
377                 inFile.close();
378                 break;
379 
380             //RE-INITIALIZE PIGS
381             case 2:
382                 // *** PIG *** /
383                 cout << "\nThe Pig(s) have been reinitialized.\n";
384                 inFile.open("PigInput.txt");
385                 getline(inFile, nameInp);
386                 inFile >> ageInp;
387                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
388                 babe.SetInitialValues(nameInp, ageInp);
389                 getline(inFile, tailInp);
390                 if (tailInp == "STRAIGHT")
391                 {
392                     tail = STRAIGHT;
393                 }
394                 else if (tailInp == "CORKSCREW")
395                 {
396                     tail = CORKSCREW;
397                 }
398                 else if (tailInp == "CURL_UP")
399                 {
400                     tail = CURL_UP;
401                 }
402                 else if (tailInp == "CURL_RIGHT")
403                 {
404                     tail = CURL_RIGHT;
405                 }
406                 else if (tailInp == "CURL_LEFT")
407                 {
408                     tail = CURL_LEFT;
409                 }
410                 babe.SetTail(tail);
411                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
412 
413                 getline(inFile, nameInp);
414                 inFile >> ageInp;
415                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
416                 wilbur.SetInitialValues(nameInp, ageInp);
417                 getline(inFile, tailInp);
418                 if (tailInp == "STRAIGHT")
419                 {
420                     tail = STRAIGHT;
421                 }
422                 else if (tailInp == "CURL_UP")
```

```
423                 {
424                     tail = CURL_UP;
425                 }
426                 else if (tailInp == "CURL_RIGHT")
427                 {
428                     tail = CURL_RIGHT;
429                 }
430                 else if (tailInp == "CURL_LEFT")
431                 {
432                     tail = CURL_LEFT;
433                 }
434                 wilbur.SetTail(tail);
435                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
436 
437                 getline(inFile, nameInp);
438                 inFile >> ageInp;
439                 inFile.ignore(numeric_limits<streamsize>::max(), '\n');
440                 kiki.SetInitialValues(nameInp, ageInp);
441                 getline(inFile, tailInp);
442                 if (tailInp == "STRAIGHT")
443                 {
444                     tail = STRAIGHT;
445                 }
446                 else if (tailInp == "CURL_UP")
447                 {
448                     tail = CURL_UP;
449                 }
450                 else if (tailInp == "CURL_RIGHT")
451                 {
452                     tail = CURL_RIGHT;
453                 }
454                 else if (tailInp == "CURL_LEFT")
455                 {
456                     tail = CURL_LEFT;
457                 }
458                 kiki.SetTail(tail);
459                 inFile.close();
460                 break;
461 
462                 //CHANGE AGE
463             case 3:
464                 do
465                 {
466                     invalid = false;
467                     cout << "\nWould you like to set the age of sheep or pigs? ↵
                      ";
468                     getline(cin, animalInp);
469                     if (animalInp != "pigs" && animalInp != "sheep")
470                     {
```

```
471                         cout << "\n**** " << animalInp << " is an invalid
                        entry ****\n";
472                         cout << "**** Please input (sheep or pigs)   ****\n";
473                         cin.clear();
474                         invalid = true;
475                     }
476
477                 } while (invalid);
478
479                 if (animalInp == "sheep")
480                 {
481                     cout << "\n1 - " << fluffy.GetName() << endl;
482                     cout << "2 - " << maa.GetName()    << endl;
483                     cout << "3 - " << lala.GetName()   << endl;
484
485                     do
486                     {
487                         invalid = false;
488                         cout << "\nSelect the animal you'd like to change: ";
489                         if (!(cin >> inp))
490                         {
491                             cout << "\n**** Please enter a NUMBER between 0
                        and 3 ****\n";
492                             cin.clear();
493                             cin.ignore(numeric_limits<streamsize>::max(),
                        '\n');
494                             invalid = true;
495                         }
496                         else if (inp < 0 || inp > 3)
497                         {
498
499                             cout << "\n**** The number " << inp << " is an
                        invalid entry     *****\n";
500                             cout << "**** Please input a number between 0 and
                        3 *****\n";
501                             invalid = true;
502                         }
503
504                     } while (invalid);
505
506                     do
507                     {
508                         invalid = false;
509                         cout << "\nNEW AGE: ";
510                         if (!(cin >> ageInp))
511                         {
512                             cout << "\n**** Please enter a NUMBER between 0
                        and 10 ****\n";
513                             cin.clear();
```

```
514                             cin.ignore(numeric_limits<streamsize>::max(),
                    '\n');
515                             invalid = true;
516                         }
517                         else if (ageInp < 0 || ageInp > 10)
518                         {
519
520                             cout << "\n**** The number " << ageInp << " is an
                    invalid entry     *****\n";
521                             cout << "**** Please input a number between 0 and
                    10 *****\n";
522                             invalid = true;
523                         }
524
525                     } while (invalid);
526
527                     switch (inp)
528                     {
529                     case 1:
530                         fluffy.ChangeAge(ageInp);
531                         cout << "Changing " << fluffy.GetName() << "\'s age to
                     " << ageInp << " ...\n";
532                         break;
533                     case 2:
534                         maa.ChangeAge(ageInp);
535                         cout << "Changing " << maa.GetName() << "\'s age to "
                    << ageInp << " ...\n";
536                         break;
537                     case 3:
538                         lala.ChangeAge(ageInp);
539                         cout << "Changing " << lala.GetName() << "\'s age to "
                     << ageInp << " ...\n";
540                         break;
541                     }
542                 }
543                 else
544                 {
545                     cout << "\n1 - " << babe.GetName() << endl;
546                     cout << "2 - " << wilbur.GetName() << endl;
547                     cout << "3 - " << kiki.GetName() << endl;
548
549                     do
550                     {
551                         invalid = false;
552                         cout << "\nSelect the animal you'd like to change: ";
553                         if (!(cin >> inp))
554                         {
555                             cout << "\n**** Please enter a NUMBER between 0
                    and 3 ****\n";
```

```
556                             cin.clear();
557                             cin.ignore(numeric_limits<streamsize>::max(),
                        '\n');
558                             invalid = true;
559                         }
560                         else if (inp < 0 || inp > 3)
561                         {
562 
563                             cout << "\n**** The number " << inp << " is an
                        invalid entry     *****\n";
564                             cout << "**** Please input a number between 0 and
                        3 *****\n";
565                             invalid = true;
566                         }
567 
568                     } while (invalid);
569 
570                     do
571                     {
572                         invalid = false;
573                         cout << "\nNEW AGE: ";
574                         if (!(cin >> ageInp))
575                         {
576                             cout << "\n**** Please enter a NUMBER between 0
                        and 10 ****\n";
577                             cin.clear();
578                             cin.ignore(numeric_limits<streamsize>::max(),
                        '\n');
579                             invalid = true;
580                         }
581                         else if (ageInp < 0 || ageInp > 10)
582                         {
583 
584                             cout << "\n**** The number " << ageInp << " is an
                        invalid entry     *****\n";
585                             cout << "**** Please input a number between 0 and
                        10 *****\n";
586                             invalid = true;
587                         }
588 
589                     } while (invalid);
590 
591                     switch (inp)
592                     {
593                     case 1:
594                         babe.ChangeAge(ageInp);
595                         cout << "Changing " << babe.GetName() << "\'s age to "
                         << ageInp << " ...\n";
596                         break;
```

```
597                     case 2:
598                         wilbur.ChangeAge(ageInp);
599                         cout << "Changing " << wilbur.GetName() << "\'s age to ↵
                        " << ageInp << " ...\n";
600                         break;
601                     case 3:
602                         kiki.ChangeAge(ageInp);
603                         cout << "Changing " << kiki.GetName() << "\'s age to " ↵
                        << ageInp << " ...\n";
604                         break;
605                     }
606                 }
607                 break;
608 
609 
610                 //DISPLAY
611             case 4:
612                 cout << "\nTHE SHEEP:\n";
613                 cout << endl;
614                 cout << left;
615                 cout << setw(NAME_SIZE) << "NAME";
616                 cout << setw(AGE_SIZE) << "AGE";
617                 cout << setw(WOOL_SIZE) << "WOOLTYPE";
618                 cout << "COLOR\n";
619                 cout << setw(NAME_SIZE) << string(NAME_SIZE - 1, '-');
620                 cout << setw(AGE_SIZE) << string(AGE_SIZE - 1, '-');
621                 cout << setw(WOOL_SIZE) << string(WOOL_SIZE - 1, '-');
622                 cout << string(5, '-') << endl;
623                 fluffy.Display();
624                 fluffy.DisplayWool();
625                 maa.Display();
626                 maa.DisplayWool();
627                 lala.Display();
628                 lala.DisplayWool();
629 
630                 cout << "\nTHE PIG(S):\n";
631                 cout << endl;
632                 cout << left;
633                 cout << setw(NAME_SIZE) << "NAME";
634                 cout << setw(AGE_SIZE)  << "AGE";
635                 cout << "TAILTYPE\n";
636                 cout << setw(NAME_SIZE) << string(NAME_SIZE - 1, '-');
637                 cout << setw(AGE_SIZE) << string(AGE_SIZE - 1, '-');
638                 cout << string(9, '-') << endl;
639                 babe.Display();
640                 babe.DisplayTail();
641                 wilbur.Display();
642                 wilbur.DisplayTail();
643                 kiki.Display();
```

```
644                     kiki.DisplayTail();
645                     break;
646                 }
647
648             } while (input != 0);
649
650
651         }
652
653
654
655         return 0;
656
657     }
```

```cpp
#include "Header.h"

/*****************************************************************************
 * PrintHeaderFile
 *   This function will output the header information
 *
 * PRE-CONDITIONS
 *   The following parameters need to have a defined value prior to calling
 *   the function
 *          asName: The name of the assignment given in the course
 *          asNum: The number of the assignment given in the course
 *          studentName: The name of the student writing the code
 *          classInfo: The course name, date, and time of the class
 *          asType: Will either output as a lab or an assignment
 *          studentID: The Identification Number of the student
 *
*****************************************************************************/

void PrintHeaderFile(ostream& output,       // IN - output datatype.
    string asName,      // IN - assignment name
    int asNum,          // IN - assignment number
    string studentName,  // IN - student's name
    string classInfo,    // IN - class that is being taken
    char asType,         // IN - assignment type
    long long studentID) // IN - student ID
{
    output << left;
    output << "*************************************************************   \n";
    output << "*   PROGRAMMED BY : " << studentName << endl;
    output << "*   " << setw(14) << "STUDENT ID " << ": " << studentID << endl;
    output << "*   " << setw(14) << "CLASS " << ": " << classInfo << endl;
    output << "*   ";

    // PROCESSING - This will adjust setws and format appropriately based
    //              on if this is a lab 'L' or assignment

    if (toupper(asType) == 'L')
    {
        output << "LAB #" << setw(9);
    }
    else
    {
        output << "ASSIGNMENT #" << setw(2);
    }
    output << asNum << ": " << asName << endl;
    output << "*************************************************************";
    output << right << endl;
```

```
47
48      return;
49  }
```

```
1  #ifndef ANIMAL_H_
2  #define ANIMAL_H_
3
4  #include "Header.h"
5
6  class Animal
7  {
8  public:
9      Animal();   // COSTRUCTOR
10     ~Animal();  // DECONSTRUCTOR
11
12     /*********************
13      **    MUTATORS    **
14      *********************/
15     void SetInitialValues(string aName,
16                           int aAge);    // Sets initialize values of the
                     animal.
17     void ChangeAge(int aAge);           // Changes the age of the animal.
18     void ChangeName(string aName);      // Cahnges the name of the animal.
19
20     /*********************
21      **    ACCESSORS    **
22      *********************/
23     void Display() const;   // Displays the name and age of the animal.
24     string GetName() const; // returns the name of the animal.
25     int GetAge() const;     // returns the age of the animal.
26
27 protected:
28     string   name;  // IN & OUT - Animal name.
29     int       age;  // IN & OUT - Animal age.
30 };
31
32 #endif
33
34
```

```cpp
#include "Header.h"
#include "Animal.h"

Animal::Animal() // CONSTRUCTOR
{
    age = 0;
}

Animal ::~Animal() {} // DESCONTRUCTOR

//MUTATORS
void Animal::SetInitialValues(string aName,
    int aAge)
{
    name = aName;
    age = aAge;
}

void Animal::ChangeAge(int aAge)
{
    age = aAge;
}

void Animal::ChangeName(string aName)
{
    name = aName;
}


//ACCESSORS
void Animal::Display() const
{
    cout << setw(NAME_SIZE) << name;
    cout  << " " << setw(AGE_SIZE - 1) << age;
}

string Animal::GetName() const
{
    return name;
}


int Animal::GetAge() const
{
    return age;
}

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

```
     **
49   * SetInitialValues
50   *   This function will set all the initial values for the animal object.
51   *
52   * INPUTS:
53   *   aName  : choice of name.
54   *   aAge   : choice of age.
55   *
56   * No outputs.
57   *                                                                        ↵
       ************************************************************************ ↵
       */
58
59   /                                                                        ↵
       ************************************************************************ ↵
       **
60   * ChangeAge
61   *   This function will set the age of the animal object.
62   *
63   * INPUTS:
64   *   aAge : selected age to change to.
65   *
66   * No outputs.
67   *                                                                        ↵
       ************************************************************************ ↵
       */
68
69   /                                                                        ↵
       ************************************************************************ ↵
       **
70   * ChangeName
71   *   This function will set the name of the animal object.
72   *
73   * INPUTS:
74   *   aName : selected name to change to.
75   *
76   * No outputs.
77   *                                                                        ↵
       ************************************************************************ ↵
       */
78
79   /                                                                        ↵
       ************************************************************************ ↵
       **
80   * Display
81   *   This function will display the object's full information.
82   *
83   * No Inputs.
84   * No outputs.
```

```
 85  *                                                                      ↵
     ************************************************************************ ↵
     */
 86
 87  /                                                                      ↵
     ************************************************************************ ↵
     **
 88  * GetName
 89  *   This function will return the name of the animal.
 90  *
 91  * No inputs.
 92  *
 93  *  OUTPUTS:
 94  *   name : Animal's name.
 95  *                                                                      ↵
     ************************************************************************ ↵
     */
 96
 97  /                                                                      ↵
     ************************************************************************ ↵
     **
 98  * GetAge
 99  *   This function will return the age of the animal.
100  *
101  * No inputs.
102  *
103  *  OUTPUTS:
104  *   age : Animal's age.
105  *                                                                      ↵
     ************************************************************************ ↵
     */
```

```
1  #ifndef SHEEP_H_
2  #define SHEEP_H_
3
4  #include "Header.h"
5  #include "Animal.h"
6  using namespace std;
7
8  enum WoolType{LONG, MEDIUM, FINE, CARPET}; // Enum for wooltypes.
9
10 class Sheep : public Animal
11 {
12 public:
13     Sheep();  // CONSTRUCTOR
14     ~Sheep(); // DECONSTRUCTOR
15
16     /*** MUTATORS ***/
17     void SetWool(WoolType wool,
18                  string woolColor); // Sets the wool type and color.
19
20     /*** ACCESSORS ***/
21     WoolType GetWool() const;     // returns the type of wool.
22     string   GetColor() const;    // returns the color of the wool.
23     void     DisplayWool() const; // Displays the wool type and color.
24
25 private:
26     WoolType wool;  // IN & OUT - Wool type.
27     string   color; // IN & OUT - Wool color.
28 };
29
30
31
32 #endif
```

```cpp
#include "Header.h"
#include "Animal.h"
#include "Sheep.h"


Sheep::Sheep(){}  // CONSTRUCTOR
Sheep::~Sheep(){} // DECONSTRUCTOR

/*** MUTATORS ***/
void Sheep::SetWool(WoolType woolType,
    string woolColor)
{
    wool = woolType;
    color = woolColor;
}

/*** ACESSORS ***/
WoolType Sheep::GetWool() const
{
    return wool;
}

string Sheep::GetColor() const
{
    return color;
}

void Sheep::DisplayWool() const
{
    string output; // CALC & OUT - sheep wooltype output.

    if (wool == LONG)
    {
        output = "Long";
    }
    else if (wool == MEDIUM)
    {
        output = "Medium";
    }
    else if (wool == FINE)
    {
        output = "Fine";
    }
    else if (wool == CARPET)
    {
        output = "Carpet";
    }
    cout << setw(WOOL_SIZE) << output << color;
    cout << endl;
```

```
50  }
51
52  /*************************************************************************
53  * SetWool
54  *   This method will set the wool type for the sheep.
55  * INPUTS:
56  *  woolType - wool type for the sheep.
57  *
58  * No outputs.
59  *                                                                       ↵
      ************************************************************************ ↵
      /
60
61  /*************************************************************************
62  * GetWool
63  *   This method will return the wool type for the sheep.
64  * No inputs.
65  *
66  * OUTPUTS:
67  *  woolType : wool type for the sheep.
68  *                                                                       ↵
      ************************************************************************ ↵
      /
69
70  /*************************************************************************
71  * GetColor
72  *   This method will return the color for the sheep.
73  * No inputs.
74  *
75  * OUTPUTS:
76  *  color : Sheep's color.
77  *                                                                       ↵
      ************************************************************************ ↵
      /
78
79  /*************************************************************************
80  * DisplayTail
81  *   This method will display the wool type and color for the sheep.
82  * No inputs.
83  * No outputs.
84  *                                                                       ↵
      ************************************************************************ ↵
      /
```

```
1  #ifndef PIG_H_
2  #define PIG_H_
3
4  #include "Header.h"
5  #include "Animal.h"
6  using namespace std;
7
8  enum TailType{STRAIGHT, CORKSCREW, CURL_UP, CURL_RIGHT, CURL_LEFT}; // Enum for
      tailtypes.
9
10 class Pig : public Animal
11 {
12 public:
13     Pig();  // CONSTRUCTOR
14     ~Pig(); // DECONSTRUCTOR
15
16     /*** MUTATORS ***/
17     void SetTail(TailType tail);  // sets the tailtype.
18
19     /*** ACCESSORS ***/
20     TailType GetTail() const;     // returns the tailtype.
21     void     DisplayTail() const; // Displays the tailtype.
22
23 private:
24     TailType tail; // IN & OUT - Tailtype.
25 };
26
27
28
29 #endif
30
```

```cpp
#include "Header.h"
#include "Animal.h"
#include "Pig.h"

/** CONSTRUCTOR & DECONSTRUCTOR **/
Pig::Pig(){}
Pig::~Pig(){}

/*** MUTATORS ***/
void Pig::SetTail(TailType tailType)
{
    tail = tailType;
}

//*** ACESSORS ***/
TailType Pig::GetTail() const
{
    return tail;
}


void Pig::DisplayTail() const
{
    string output; // CALC & OUT - Output tail.

    if (tail == STRAIGHT)
    {
        output = "Straight";
    }
    else if (tail == CORKSCREW)
    {
        output = "Corkscrew";
    }
    else if (tail == CURL_UP)
    {
        output = "Curl Up";
    }
    else if (tail == CURL_RIGHT)
    {
        output = "Curl Right";
    }
    else if (tail == CURL_LEFT)
    {
        output = "Curl Left";
    }

    cout << output;
    cout << endl;
}
```

```
50
51  /*****************************************************************************
52  * SetTail
53  *   This method will set the tail type for the pig.
54  * INPUTS:
55  *  tailType - tail type of pig.
56  *
57  * No outputs.
58  *                                                                            ↵
      ***************************************************************************** ↵
      /
59
60  /*****************************************************************************
61  * GetTail
62  *   This method will return the tail type for the pig.
63  * No inputs.
64  *
65  * OUTPUTS:
66  *  tailtype : tail type of pig.
67  *                                                                            ↵
      ***************************************************************************** ↵
      /
68
69  /*****************************************************************************
70  * DisplayTail
71  *   This method will display the tail type for the pig.
72  * No inputs.
73  * No outputs.
74  *                                                                            ↵
      ***************************************************************************** ↵
      /
```