```
 1 ************************************************
 2 *  PROGRAMMED BY : Daniel Olaes & Faris Hijazi
 3 *  STUDENT ID    : 1128957 & 1039438
 4 *  CLASS         : CS1B - MW - 7:30pm
 5 *  LAB #12       : Intro to OOP
 6 ************************************************
 7
 8 MENU
 9 1 - Initialize Animals
10 0 - EXIT
11 Enter Selection: 2
12
13 **** The number 2 is an invalid entry      ****
14 **** Please input a number between 0 and 1 ****
15
16 MENU
17 1 - Initialize Animals
18 0 - EXIT
19 Enter Selection: a
20
21 **** Please input a NUMBER between 0 and 1 ****
22
23 MENU
24 1 - Initialize Animals
25 0 - EXIT
26 Enter Selection: 1
27
28 Initializing Fluffy, Maa, Babe...
29
30 MENU
31 1 - Initialize Animals
32 2 - Change Age
33 3 - Change Value
34 4 - Display
35 0 - EXIT
36 Enter Selection: 5
37
38 **** The number 5 is an invalid entry      ****
39 **** Please input a number between 0 and 4 ****
40
41 MENU
42 1 - Initialize Animals
43 2 - Change Age
44 3 - Change Value
45 4 - Display
46 0 - EXIT
47 Enter Selection: 4
48
49 ANIMAL     NAME           AGE    VALUE
50 ---------- -------------- --- -----------
51 Sheep      Fluffy           1    15000.00
52 Sheep      Maa              3    16520.35
53 Pig        Babe             2    10240.67
54
55 MENU
56 1 - Initialize Animals
57 2 - Change Age
58 3 - Change Value
59 4 - Display
60 0 - EXIT
61 Enter Selection: 2
62
63 CHANGE AGE:
64 1 - Fluffy
```

```
2 - Maa
3 - Babe
Select the animal you'd like to choose: 4

**** The number 4 is an invalid entry      ****
**** Please input a number between 1 and 3 ****

CHANGE AGE:
1 - Fluffy
2 - Maa
3 - Babe
Select the animal you'd like to choose: a

**** Please input a NUMBER between 1 and 3 ****

CHANGE AGE:
1 - Fluffy
2 - Maa
3 - Babe
Select the animal you'd like to choose: 1

NEW AGE: 2

Changing Fluffy's age to 2...

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 2

CHANGE AGE:
1 - Fluffy
2 - Maa
3 - Babe
Select the animal you'd like to choose: 2

NEW AGE: 4

Changing Maa's age to 4...

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 2

CHANGE AGE:
1 - Fluffy
2 - Maa
3 - Babe
Select the animal you'd like to choose: 3

NEW AGE: 11

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

NEW AGE: 3

```

```
Changing Babe's age to 3...

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 3

CHANGE VALUE:
1 - Fluffy
2 - Maa
3 - Babe
3 - Babe
Select the animal you'd like to choose: 4

**** The number 4 is an invalid entry      ****
**** Please input a number between 1 and 3 ****

CHANGE VALUE:
1 - Fluffy
2 - Maa
3 - Babe
Select the animal you'd like to choose: 1

NEW VALUE: 154154.51

Changing Fluffy's value to 154154.52...

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 3

CHANGE VALUE:
1 - Fluffy
2 - Maa
3 - Babe
Select the animal you'd like to choose: 2

NEW VALUE: 651651.61

**** The number 651652 is an invalid entry      ****
**** Please input a number between 0 and 400000 ****

NEW VALUE: 165165.61

Changing Maa's value to 165165.61...

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 3

CHANGE VALUE:
1 - Fluffy
2 - Maa
```

```
3 - Babe
Select the animal you'd like to choose: 3

NEW VALUE: 123123.12

Changing Babe's value to 123123.12...

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 4

ANIMAL     NAME           AGE    VALUE
---------- -------------- --- -----------
Sheep      Fluffy           2   154154.52
Sheep      Maa              4   165165.61
Pig        Babe             3   123123.12

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 1

Are you sure you want to reinitialize (Y/N)? x

**** x is an invalid entry ****
**** Please Enter Y or N   ****

Are you sure you want to reinitialize (Y/N)? n
Animals have not been reinitialized!

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 4

ANIMAL     NAME           AGE    VALUE
---------- -------------- --- -----------
Sheep      Fluffy           2   154154.52
Sheep      Maa              4   165165.61
Pig        Babe             3   123123.12

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 1

Are you sure you want to reinitialize (Y/N)? Y
Initializing Fluffy, Maa, Babe...

MENU
1 - Initialize Animals
```

```
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 4

ANIMAL     NAME           AGE    VALUE
---------- -------------- --- -----------
Sheep      Fluffy           1    15000.00
Sheep      Maa              3    16520.35
Pig        Babe             2    10240.67

MENU
1 - Initialize Animals
2 - Change Age
3 - Change Value
4 - Display
0 - EXIT
Enter Selection: 0
```

```
/*****************************************************************************
 * AUTHOR     :Daniel Olaes & Faris Hijazi
 * STUDENT ID :1128957 & 1039438
 * LAB #12    :Intro to OOP
 * CLASS      :CS1B
 * SECTION    :MW: 7:30pm
 * DUE DATE   :4/25/19
 *****************************************************************************/

#ifndef CLASSHEADER_H_
#define CLASSHEADER_H_

#include <iomanip>
#include <string>
using namespace std;

class Animal
{
    public:
        Animal();
        ~Animal();

        void SetInitialValues(string aName, string aType, int aAge,
                              float aValue);
        void ChangeAge(int aAge);
        void ChangeValue(float aValue);

        void   Display() const;
        string GetName() const;
        string GetType() const;
        int    GetAge() const;
        float  GetValue() const;

    private:
        string name;
        string type;
        int    age;
        float  value;

};

#endif /* CLASSHEADER_H_ */

```

```
 1 /*******************************************************************************
 2  * AUTHOR     :Daniel Olaes & Faris Hijazi
 3  * STUDENT ID :1128957 &
 4  * LAB #12    :Intro to OOP
 5  * CLASS      :CS1B
 6  * SECTION    :MW: 7:30pm
 7  * DUE DATE   :4/25/19
 8  ******************************************************************************/
 9
10 #ifndef MAINHEADER_H_
11 #define MAINHEADER_H_
12
13 /*******************************************************************************
14  * PREPROCESSOR DIRECTIVES
15  ******************************************************************************/
16 #include <iostream>
17 #include <iomanip>
18 #include <string>
19 #include <limits>
20 #include <ios>
21 //#include <stdio.h>
22 #include <math.h>
23 #include "ClassHeader.h"
24 using namespace std;
25
26 /*******************************************************************************
27  * ENUMERATED TYPES
28  ******************************************************************************/
29 enum MainMenu
30 {
31     Exit,
32     InitializeAnimals,
33     AgeChange,
34     ValueChange,
35     DisplayAnimals
36 };
37
38 enum AnimalList
39 {
40     AnimalOne,
41     AnimalTwo,
42     AnimalThree
43 };
44
45 /*******************************************************************************
46  * FUNCTION PROTOTYPES
47  ******************************************************************************/
48
49 /*******************************************************************************
50  * ERROR CHECKING PROTOTYPES
51  ******************************************************************************/
52
53 /*******************************************************************************
54  * FUNCTION InputMenuOp
55  *------------------------------------------------------------------------------
56  * This function allows the user to select a menu option and validate the menu
57  * option inputed. If menuType is 'S', then the small menu will be output and
58  * the user input validated according to the small menu. If menuType is 'M', the
59  * main menu will be output and the user input validated according to the main
60  * menu.
61  *
62  * RETURN: the validated menu options selected by the user
63  ******************************************************************************/
64 int InputMenuOp(char menuType); //IN - the type of menu option
```

```
 65
 66 /*****************************************************************************
 67  * FUNCTION InputAnimalSelected
 68  *----------------------------------------------------------------------------
 69  * This function allows the user to select a animal menu option and validate the
 70  * animal menu option inputed.
 71  *
 72  * RETURN: the validated animal menu options selected by the user
 73  *****************************************************************************/
 74 int InputAnimalSelected(char headerType);
 75
 76 /*****************************************************************************
 77  * FUNCTION InputNewAge
 78  *----------------------------------------------------------------------------
 79  * This function allows the user to enter a new age and validate that the age is
 80  * within the specific range of 0 to 10 years old
 81  *
 82  * RETURN: the validated new age entered by the user
 83  *****************************************************************************/
 84 int InputNewAge();
 85
 86 /*****************************************************************************
 87  * FUNCTION InputNewValue
 88  *----------------------------------------------------------------------------
 89  * This function allows the user to enter a new value and validate that the
 90  * value is within the specific range of 0 to 400000
 91  *
 92  * RETURN: the validated new value entered by the user
 93  *****************************************************************************/
 94 float InputNewValue();
 95
 96 /*****************************************************************************
 97  * FUNCTION ConfirmReinitialize
 98  *----------------------------------------------------------------------------
 99  * This function allows the user to select 'Y' or 'N' to confirm if they wish to
100  * reinitialize the animals, and validates their request.
101  *
102  * RETURN: the validated answer entered by the user
103  *****************************************************************************/
104 char ConfirmReinitialize();
105
106 /*****************************************************************************
107  * MISC FUNCTIONS
108  *****************************************************************************/
109
110 /*****************************************************************************
111  * FUNCTION PrintHeaderOStream
112  *----------------------------------------------------------------------------
113  * This function receives an output type, assignment name, type and number then
114  * outputs appropriate header to the corresponding location - return nothing.
115  *
116  * RETURNS: nothing
117  *  - the class header is output onto the screen IO.
118  *****************************************************************************/
119
120 void PrintHeaderOStream ( ostream &output,  // IN - the type of output statement
121                           string  asName,   // IN - assignment name
122                           int     asNum,    // IN - assignment number
123                           char    asType ); // IN - assignment type
124                                             //      ('L' = Lab
125                                             //       'A' = Assignment)
126
127 /*****************************************************************************
128  * FUNCTION DisplayAnimalHeader
```

```
129 *------------------------------------------------------------------------------
130 * This function will output the the animal list header to the screen IO.
131 *
132 * RETURN: nothing
133 *  - the animal list header is printed to the screen IO
134 ******************************************************************************/
135 void DisplayAnimalHeader();
136
137 #endif /* MAINHEADER_H_ */
138
```

```cpp
/*******************************************************************************
 * AUTHOR     :Daniel Olaes & Faris Hijazi
 * STUDENT ID :1128957 & 1039438
 * LAB #12    :Intro to OOP
 * CLASS      :CS1B
 * SECTION    :MW: 7:30pm
 * DUE DATE   :4/25/19
 ******************************************************************************/

#include "MainHeader.h"

/*******************************************************************************
 * LAB 12 - INTRO TO OOP
 *------------------------------------------------------------------------------
 *
 ******************************************************************************/

int main()
{
    /*******************************************************************************
     * VARIABLES
     ******************************************************************************/
    Animal fluffy; //CALC - instance of animal class
    Animal maa;    //CALC - instance of animal class
    Animal babe;   //CALC - instance of animal class

    int    menuOption;   //IN - menu option input by user
    char   initialCon;   //IN - gets initial conditions for initialization
    int    animalOption; //IN - gets animal option user selected
    int    newAge;       //IN - value of new age input by user
    float  newValue;     //IN - new animal value input by user

    PrintHeaderOStream ( cout, "Intro to OOP", 12, 'L');

    menuOption = InputMenuOp('S');

    if(menuOption == InitializeAnimals)
    {
        cout << "\nInitializing Fluffy, Maa, Babe...\n";
        fluffy.SetInitialValues("Fluffy", "Sheep", 1, 15000.00);
        maa.SetInitialValues("Maa", "Sheep", 3, 16520.35);
        babe.SetInitialValues("Babe", "Pig", 2, 10240.67);

        menuOption = InputMenuOp('M');

        while(menuOption != 0)
        {

            switch(menuOption)
            {
                case InitializeAnimals:initialCon = ConfirmReinitialize();
                    if(initialCon == 'Y')
                    {
                        cout << "Initializing Fluffy, Maa, "
                            << "Babe...\n";
                        fluffy.SetInitialValues("Fluffy",
                                                "Sheep",
                                                1,
                                                15000.00);
                        maa.SetInitialValues("Maa",
                                             "Sheep",
                                                3,
                                                16520.35);
                        babe.SetInitialValues("Babe",
```

```
                                        "Pig",
                                        2,
                                        10240.67);
                }
                else
                {
                    cout << "Animals have not been "
                        << "reinitialized!\n";
                }
                break;
            case AgeChange:animalOption = InputAnimalSelected('A');
                newAge = InputNewAge();
                cout << "\nChanging ";
                switch(animalOption)
                {
                    case AnimalOne:cout << "Fluffy's ";
                                   fluffy.ChangeAge(newAge);
                                   break;
                    case AnimalTwo:cout << "Maa's ";
                                   maa.ChangeAge(newAge);
                                   break;
                    case AnimalThree:cout << "Babe's ";
                                   babe.ChangeAge(newAge);
                                   break;
                }
                cout << "age to " << newAge << "...\n";
                break;

            case ValueChange:animalOption = InputAnimalSelected('V');
                newValue = InputNewValue();
                cout << "\nChanging ";
                switch(animalOption)
                {
                    case AnimalOne:cout << "Fluffy's ";
                                fluffy.ChangeValue(newValue);
                                break;
                    case AnimalTwo:cout << "Maa's ";
                                maa.ChangeValue(newValue);
                                break;
                    case AnimalThree:cout << "Babe's ";
                                babe.ChangeValue(newValue);
                                break;
                }
                cout << fixed << setprecision(2);
                cout << "value to " << newValue << "...\n";
                cout.unsetf(ios::fixed);
                cout << setprecision(6);
                break;

            case DisplayAnimals:DisplayAnimalHeader();
                fluffy.Display();
                maa.Display();
                babe.Display();
                break;
            case Exit: break;
        }

        menuOption = InputMenuOp('M');

    }
}
}
```

```cpp
/*****************************************************************************
 * AUTHOR     :Daniel Olaes & Faris Hijazi
 * STUDENT ID :1128957 & 1039438
 * LAB #12    :Intro to OOP
 * CLASS      :CS1B
 * SECTION    :MW: 7:30pm
 * DUE DATE   :4/25/19
 *****************************************************************************/

#include "MainHeader.h"
#include "ClassHeader.h"

/*****************************************************************************
 * CONSTRUCTOR Animal
 *----------------------------------------------------------------------------
 *
 * RETURN: NA
 *****************************************************************************/
Animal::Animal()
{
    name.clear();
    type.clear();
    age   = 0;
    value = 0;
}

//----------------------------------------------------------------------------

/*****************************************************************************
 * DECONSTRUCTOR Animal
 *----------------------------------------------------------------------------
 *
 * RETURN: NA
 *****************************************************************************/
Animal::~Animal()
{
    //NOTHING
}

//----------------------------------------------------------------------------

/*****************************************************************************
 * METHOD SetInitialValues
 *----------------------------------------------------------------------------
 * sets the initial values for an object of class animal.
 * RETURN: NA
 *****************************************************************************/
void Animal::SetInitialValues(string aName,
                              string aType,
                              int    aAge,
                              float  aValue)
{
    name  = aName;
    type  = aType;
    age   = aAge;
    value = aValue;
}

//----------------------------------------------------------------------------

/*****************************************************************************
 * METHOD ChangeAge
 *----------------------------------------------------------------------------
 * changes the age of an object of class animal.
```

```
 * RETURN: NA
 ******************************************************************************/
void Animal::ChangeAge(int aAge)
{
    age = aAge;
}

//-----------------------------------------------------------------------------

/******************************************************************************
 * METHOD ChangeValue
 *-----------------------------------------------------------------------------
 * changes the value of an object of class animal.
 * RETURN: NA
 ******************************************************************************/
void Animal::ChangeValue(float aValue)
{
    value = aValue;
}

//-----------------------------------------------------------------------------

/******************************************************************************
 * METHOD Display
 *-----------------------------------------------------------------------------
 * displays all atributes of an object of class animal
 * RETURN: NA
 ******************************************************************************/
void Animal::Display() const
{
    /**************************************************************************
     * CONSTANT
     *-------------------------------------------------------------------------
     * ANIMAL_COL - size of animal col
     * NAME_COL   - size of name col
     * AGE_COL    - size of age col
     * VALUE_COL  - size of value col
     **************************************************************************/
    const int ANIMAL_COL = 11;
    const int NAME_COL   = 15;
    const int AGE_COL    = 3;
    const int VALUE_COL  = 12;

    cout << left;
    cout << setw(ANIMAL_COL) << type;
    cout << setw(NAME_COL)   << name;
    cout << right;
    cout << setw(AGE_COL)    << age;
    cout << fixed << setprecision(2);
    cout << setw(VALUE_COL)  << value << endl;
    cout.unsetf(ios::fixed);
    cout << setprecision(6);
}

//-----------------------------------------------------------------------------

/******************************************************************************
 * METHOD GetName
 *-----------------------------------------------------------------------------
 * returns name atribute.
 * RETURN: name
 ******************************************************************************/
string Animal::GetName() const
{
```

```cpp
    return name;
}

//---------------------------------------------------------------------------

/*****************************************************************************
 * METHOD GetType
 *---------------------------------------------------------------------------
 * returns type atribute.
 * RETURN: type
 *****************************************************************************/
string Animal::GetType() const
{
    return type;
}

//---------------------------------------------------------------------------

/*****************************************************************************
 * METHOD GetAge
 *---------------------------------------------------------------------------
 * returns age atribute.
 * RETURN: age
 *****************************************************************************/
int Animal::GetAge() const
{
    return age;
}

//---------------------------------------------------------------------------

/*****************************************************************************
 * METHOD GetValue
 *---------------------------------------------------------------------------
 * returns value atribute.
 * RETURN: value
 *****************************************************************************/
float Animal::GetValue() const
{
    return value;
}

```

```cpp
/*******************************************************************************
 * AUTHOR     :Daniel Olaes & Faris Hijazi
 * STUDENT ID :1128957 & 1039438
 * LAB #12    :Intro to OOP
 * CLASS      :CS1B
 * SECTION    :MW: 7:30pm
 * DUE DATE   :4/25/19
 ******************************************************************************/

#include "MainHeader.h"

/*******************************************************************************
 * FUNCTION InputMenuOp
 *------------------------------------------------------------------------------
 * This function allows the user to select a menu option and validate the menu
 * option inputed. If menuType is 'S', then the small menu will be output and
 * the user input validated according to the small menu. If menuType is 'M', the
 * main menu will be output and the user input validated according to the main
 * menu.
 *
 * RETURN: the validated menu options selected by the user
 ******************************************************************************/

int InputMenuOp(char menuType) //IN - the type of menu option
{
    /*******************************************************************************
     * CONSTANTS
     *------------------------------------------------------------------------------
     * ERROR_MESSAGE - the number of spaces reserved for the error message
     ******************************************************************************/
    const int ERROR_MESSAGE = 43;

    /*******************************************************************************
     * VARIABLES
     ******************************************************************************/
    int    menuOp;    //IN,CALC,OUT - the menu option selected by the user
    string menuError; //CALC        - the string that holds the error message
    bool   menuValid; //CALC        - the boolean that checks the validity of
                      //            - the list
    if(menuType == 'S')
    {
        do
        {
            cout << "MENU\n"
                 << "1 - Initialize Animals\n"
                 << "0 - EXIT\n"
                 << "Enter Selection: ";

            menuValid = false;

            cout << left;

            if(!(cin >> menuOp))
            {
                cout << setw(ERROR_MESSAGE)
                     << "\n**** Please input a NUMBER between 0 and 1 ****"
                     << endl << endl;
                cin.clear();

                cin.ignore(numeric_limits<streamsize>::max(), '\n');
            }
            else if(menuOp < 0 || menuOp > 1)
            {
                menuError = "**** The number "
```

```
65                          + to_string(menuOp)
66                          + " is an invalid entry ";
67
68                  cout << endl
69                       << setw(ERROR_MESSAGE)
70                       << menuError
71                       << "****\n";
72                  cout << setw(ERROR_MESSAGE)
73                       << "**** Please input a number between 0 and 1 ****\n\n";
74
75              }
76              else
77              {
78                  menuValid = true;
79              }
80
81          } while(!menuValid);
82
83      }
84      else if (menuType == 'M')
85      {
86
87          do
88          {
89              cout << "\nMENU\n"
90                   << "1 - Initialize Animals\n"
91                   << "2 - Change Age\n"
92                   << "3 - Change Value\n"
93                   << "4 - Display\n"
94                   << "0 - EXIT\n"
95                   << "Enter Selection: ";
96
97              menuValid = false;
98
99              cout << left;
100
101             if(!(cin >> menuOp))
102             {
103                 cout << setw(ERROR_MESSAGE)
104                      << "\n**** Please input a NUMBER between 0 and 4 ****"
105                      << endl;
106                 cin.clear();
107
108                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
109             }
110             else if(menuOp < 0 || menuOp > 4)
111             {
112                 menuError = "**** The number "
113                             + to_string(menuOp)
114                             + " is an invalid entry ";
115
116                 cout << endl
117                      << setw(ERROR_MESSAGE)
118                      << menuError
119                      << "****\n";
120                 cout << setw(ERROR_MESSAGE)
121                      << "**** Please input a number between 0 and 4 ****\n";
122
123             }
124             else
125             {
126                 menuValid = true;
127             }
128
```

```
129         } while(!menuValid);
130     }
131 
132     cout << right;
133 
134     cin.ignore(1000,'\n');
135 
136     return menuOp;
137 }
138 
139 //-----------------------------------------------------------------------------
140 
141 /*****************************************************************************
142  * FUNCTION InputAnimalSelected
143  *----------------------------------------------------------------------------
144  * This function allows the user to select a animal menu option and validate the
145  * animal menu option inputed. Depending on the header type, the animal menu
146  * will have a different header for age or value change.
147  *
148  * RETURN: the validated animal menu options selected by the user
149  *****************************************************************************/
150 
151 int InputAnimalSelected(char headerType)
152 {
153     /*************************************************************************
154      * CONSTANTS
155      *------------------------------------------------------------------------
156      * ERROR_MESSAGE - the number of spaces reserved for the error message
157      *************************************************************************/
158     const int ERROR_MESSAGE = 43;
159 
160     /*************************************************************************
161      * VARIABLES
162      *************************************************************************/
163     int    animalOp;    //IN,CALC,OUT - the menu option selected by the user
164     string animalError; //CALC        - the string that holds the error message
165     bool   animalValid; //CALC        - the boolean that checks the validity of
166                         //            - the list
167 
168     do
169         {
170             if(headerType == 'A')
171             {
172                 cout << "\nCHANGE AGE:\n";
173             }
174             else if (headerType == 'V')
175             {
176                 cout << "\nCHANGE VALUE:\n";
177             }
178 
179             cout << "1 - Fluffy\n"
180                  << "2 - Maa\n"
181                  << "3 - Babe\n"
182                  << "Select the animal you'd like to choose: ";
183 
184             animalValid = false;
185 
186             cout << left;
187 
188             if(!(cin >> animalOp))
189             {
190                 cout << setw(ERROR_MESSAGE)
191                      << "\n**** Please input a NUMBER between 1 and 3 ****\n";
192                 cin.clear();
```

```
193
194                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
195             }
196             else if(animalOp < 1 || animalOp > 3)
197             {
198                 animalError = "**** The number "
199                             + to_string(animalOp)
200                             + " is an invalid entry ";
201
202                 cout << endl
203                      << setw(ERROR_MESSAGE)
204                      << animalError
205                      << "****\n";
206                 cout << setw(ERROR_MESSAGE)
207                      << "**** Please input a number between 1 and 3 ****\n";
208
209             }
210             else
211             {
212                 animalValid = true;
213             }
214
215         } while(!animalValid);
216
217     cout << right;
218
219     cin.ignore(1000,'\n');
220
221     return animalOp - 1;
222 }
223
224 //----------------------------------------------------------------------------
225
226 /*****************************************************************************
227  * FUNCTION InputNewAge
228  *----------------------------------------------------------------------------
229  * This function allows the user to enter a new age and validate that the age is
230  * within the specific range of 0 to 10 years old
231  *
232  * RETURN: the validated new age entered by the user
233  *****************************************************************************/
234
235 int InputNewAge()
236 {
237     /*************************************************************************
238      * CONSTANTS
239      *------------------------------------------------------------------------
240      * ERROR_MESSAGE - the number of spaces reserved for the error message
241      *************************************************************************/
242     const int ERROR_MESSAGE = 44;
243
244     /*************************************************************************
245      * VARIABLES
246      *************************************************************************/
247     int    age;      //IN,CALC,OUT - the new age entered by the user
248     string ageError; //CALC        - the string that holds the error message
249     bool   ageValid; //CALC        - the boolean that checks the validity of
250                      //            - the list
251
252     do
253         {
254             cout << "\nNEW AGE: ";
255
256             ageValid = false;
```

```
257
258             cout << left;
259
260             if(!(cin >> age))
261             {
262                 cout << setw(ERROR_MESSAGE)
263                      << "\n**** Please input a NUMBER between 0 and 10 ****\n";
264                 cin.clear();
265
266                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
267             }
268             else if(age < 0 || age > 10)
269             {
270                 ageError = "**** The number "
271                            + to_string(age)
272                            + " is an invalid entry ";
273
274                 cout << endl
275                      << setw(ERROR_MESSAGE)
276                      << ageError
277                      << "****\n";
278                 cout << setw(ERROR_MESSAGE)
279                      << "**** Please input a number between 0 and 10 ****\n";
280
281             }
282             else
283             {
284                 ageValid = true;
285             }
286
287         } while(!ageValid);
288
289     cout << right;
290
291     cin.ignore(1000,'\n');
292
293     return age;
294 }
295
296 //-----------------------------------------------------------------------------
297
298 /*****************************************************************************
299  * FUNCTION InputNewValue
300  *----------------------------------------------------------------------------
301  * This function allows the user to enter a new value and validate that the
302  * value is within the specific range of 0 to 400000
303  *
304  * RETURN: the validated new value entered by the user
305  *****************************************************************************/
306
307 float InputNewValue()
308 {
309     /*************************************************************************
310      * CONSTANTS
311      *------------------------------------------------------------------------
312      * ERROR_MESSAGE - the number of spaces reserved for the error message
313      *************************************************************************/
314     const int ERROR_MESSAGE = 48;
315
316     /*************************************************************************
317      * VARIABLES
318      *************************************************************************/
319     float  value;      //IN,CALC,OUT - the new value by the user
320     string valueError; //CALC        - the string that holds the error message
```

```
321     bool   valueValid; //CALC        - the boolean that checks the validity of
322                        //            - the list
323
324     do
325         {
326             cout << "\nNEW VALUE: ";
327
328             valueValid = false;
329
330             cout << left;
331
332             if(!(cin >> value))
333             {
334                 cout << setw(ERROR_MESSAGE)
335                      << "\n**** Please input a NUMBER between 0 and 400000 ****"
336                      << endl;
337                 cin.clear();
338
339                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
340             }
341             else if(value < 0 || value > 400000)
342             {
343
344                 valueError = "**** The number "
345                              + to_string(int(round(value)))
346                              + " is an invalid entry ";
347
348                 cout << endl
349                      << setw(ERROR_MESSAGE)
350                      << valueError
351                      << "****\n";
352                 cout << setw(ERROR_MESSAGE)
353                      << "**** Please input a number between 0 and 400000 ****"
354                      << endl;
355
356             }
357             else
358             {
359                 valueValid = true;
360             }
361
362         } while(!valueValid);
363
364     cout << right;
365
366     cin.ignore(1000,'\n');
367
368     return value;
369 }
370
371 //----------------------------------------------------------------------------
372
373 /******************************************************************************
374  * FUNCTION ConfirmReinitialize
375  *----------------------------------------------------------------------------
376  * This function allows the user to select 'Y' or 'N' to confirm if they wish to
377  * reinitialize the animals, and validates their request.
378  *
379  * RETURN: the validated answer entered by the user
380  *****************************************************************************/
381
382 char ConfirmReinitialize()
383 {
384     /*************************************************************************
```

```
385      * CONSTANTS
386      *--------------------------------------------------------------------------
387      * ERROR_MESSAGE - the number of spaces reserved for the error message
388      ***************************************************************************/
389     const int ERROR_MESSAGE = 27;
390
391     /***************************************************************************
392      * VARIABLES
393      ***************************************************************************/
394     char   answer;      //IN,CALC,OUT - the answer entered by the user
395     string ansString;
396     string answerError; //CALC        - the string that holds the error message
397     bool   answerValid; //CALC        - the boolean that checks the validity of
398                         //            - the list
399
400     do
401         {
402
403             answerValid = false;
404
405             cout << left;
406
407             cout << "\nAre you sure you want to reinitialize (Y/N)? ";
408             cin.get(answer);
409             cin.ignore(1000,'\n');
410
411             if(toupper(answer) != 'Y' && toupper(answer) != 'N')
412             {
413                 ansString.assign(1,answer);
414                 answerError = "**** " + ansString
415                               + " is an invalid entry ";
416
417                 cout << endl
418                      << setw(ERROR_MESSAGE)
419                      << answerError
420                      << "****";
421                 cout << setw(ERROR_MESSAGE)
422                      << "\n**** Please Enter Y or N   ****\n";
423
424             }
425             else
426             {
427                 answerValid = true;
428             }
429
430         } while(!answerValid);
431
432     cout << right;
433
434     return toupper(answer);
435 }
436
```

```cpp
/*****************************************************************************
 * AUTHOR     :Daniel Olaes & Faris Hijazi
 * STUDENT ID :1128957 & 1039438
 * LAB #12    :Intro to OOP
 * CLASS      :CS1B
 * SECTION    :MW: 7:30pm
 * DUE DATE   :4/25/19
 *****************************************************************************/

#include "MainHeader.h"
#include "ClassHeader.h"

/*****************************************************************************
 * FUNCTION PrintHeaderOStream
 *----------------------------------------------------------------------------
 * This function receives an output type, assignment name, type and number then
 * outputs appropriate header to the corresponding location - return nothing.
 *
 * RETURNS: nothing
 *  - the class header is output onto the screen IO.
 *****************************************************************************/

void PrintHeaderOStream ( ostream &output, // IN - the type of output statement
                          string  asName,  // IN - assignment name
                          int     asNum,   // IN - assignment number
                          char    asType ) // IN - assignment type
                                           //      ('L' = Lab
                                           //       'A' = Assignment)
{
    output << left;
    output << "************************************************\n";
    output << "*  PROGRAMMED BY : Daniel Olaes & Faris Hijazi\n";
    output << "*  "   << setw(14) << "STUDENT ID" << ": 1128957 & 1039438\n";
    output << "*  "   << setw(14) << "CLASS"    << ": CS1B - MW - 7:30pm\n";
    output << "*  ";

    if (toupper(asType) == 'L')
    {
        output << "LAB #" << setw(9);
    }
    else
    {
        output << "ASSIGNMENT #" << setw(2);
    }

    output << asNum << ": " << asName << endl;
    output << "************************************************\n\n";
    output << right;
}

//----------------------------------------------------------------------------

/*****************************************************************************
 * FUNCTION DisplayAnimalHeader
 *----------------------------------------------------------------------------
 * This function will output the the animal list header to the screen IO.
 *
 * RETURN: nothing
 *  - the animal list header is printed to the screen IO
 *****************************************************************************/
void DisplayAnimalHeader()
{
    /*************************************************************************
     * CONSTANT
```

```
65      *-------------------------------------------------------------------------
66      * ANIMAL_COL - animal column width
67      * NAME_COL   - name column width
68      * AGE_COL    - age column width
69      * VALUE_COL  - value column width
70      *************************************************************************/
71     const int ANIMAL_COL = 11;
72     const int NAME_COL   = 15;
73     const int AGE_COL    = 3;
74     const int VALUE_COL  = 11;
75
76     cout << left << endl
77          << setw(ANIMAL_COL)  << "ANIMAL"
78          << setw(NAME_COL)    << "NAME"
79          << right
80          << setw(AGE_COL)     << "AGE"
81          << setw(1)           << " "
82          << setw(VALUE_COL-8) << " " << "VALUE" << endl;
83
84     cout << left
85          << setw(ANIMAL_COL) << "----------"
86          << setw(NAME_COL)   << "---------------"
87          << right
88          << setw(AGE_COL)    << "---"
89          << setw(1)          << " "
90          << setw(VALUE_COL)  << "-----------\n";
91
92     cout << left;
93
94 }
95
```