

output.txt

```

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

```

output.txt

```
65 2 - Maa
66 3 - Babe
67 Select the animal you'd like to choose: 4
68
69 **** The number 4 is an invalid entry      ****
70 **** Please input a number between 1 and 3 ****
71
72 CHANGE AGE:
73 1 - Fluffy
74 2 - Maa
75 3 - Babe
76 Select the animal you'd like to choose: a
77
78 **** Please input a NUMBER between 1 and 3 ****
79
80 CHANGE AGE:
81 1 - Fluffy
82 2 - Maa
83 3 - Babe
84 Select the animal you'd like to choose: 1
85
86 NEW AGE: 2
87
88 Changing Fluffy's age to 2...
89
90 MENU
91 1 - Initialize Animals
92 2 - Change Age
93 3 - Change Value
94 4 - Display
95 0 - EXIT
96 Enter Selection: 2
97
98 CHANGE AGE:
99 1 - Fluffy
100 2 - Maa
101 3 - Babe
102 Select the animal you'd like to choose: 2
103
104 NEW AGE: 4
105
106 Changing Maa's age to 4...
107
108 MENU
109 1 - Initialize Animals
110 2 - Change Age
111 3 - Change Value
112 4 - Display
113 0 - EXIT
114 Enter Selection: 2
115
116 CHANGE AGE:
117 1 - Fluffy
118 2 - Maa
119 3 - Babe
120 Select the animal you'd like to choose: 3
121
122 NEW AGE: 11
123
124 **** The number 11 is an invalid entry      ****
125 **** Please input a number between 0 and 10 ****
126
127 NEW AGE: 3
128
```

output.txt

```
129 Changing Babe's age to 3...
130
131 MENU
132 1 - Initialize Animals
133 2 - Change Age
134 3 - Change Value
135 4 - Display
136 0 - EXIT
137 Enter Selection: 3
138
139 CHANGE VALUE:
140 1 - Fluffy
141 2 - Maa
142 3 - Babe
143 3 - Babe
144 Select the animal you'd like to choose: 4
145
146 **** The number 4 is an invalid entry ****
147 **** Please input a number between 1 and 3 ****
148
149 CHANGE VALUE:
150 1 - Fluffy
151 2 - Maa
152 3 - Babe
153 Select the animal you'd like to choose: 1
154
155 NEW VALUE: 154154.51
156
157 Changing Fluffy's value to 154154.52...
158
159 MENU
160 1 - Initialize Animals
161 2 - Change Age
162 3 - Change Value
163 4 - Display
164 0 - EXIT
165 Enter Selection: 3
166
167 CHANGE VALUE:
168 1 - Fluffy
169 2 - Maa
170 3 - Babe
171 Select the animal you'd like to choose: 2
172
173 NEW VALUE: 651651.61
174
175 **** The number 651652 is an invalid entry ****
176 **** Please input a number between 0 and 400000 ****
177
178 NEW VALUE: 165165.61
179
180 Changing Maa's value to 165165.61...
181
182 MENU
183 1 - Initialize Animals
184 2 - Change Age
185 3 - Change Value
186 4 - Display
187 0 - EXIT
188 Enter Selection: 3
189
190 CHANGE VALUE:
191 1 - Fluffy
192 2 - Maa
```

```

1933 - Babe
194Select the animal you'd like to choose: 3
195
196NEW VALUE: 123123.12
197
198Changing Babe's value to 123123.12...
199
200MENU
2011 - Initialize Animals
2022 - Change Age
2033 - Change Value
2044 - Display
2050 - EXIT
206Enter Selection: 4
207
208ANIMAL      NAME      AGE      VALUE
209-----
210Sheep      Fluffy      2      154154.52
211Sheep      Maa         4      165165.61
212Pig        Babe         3      123123.12
213
214MENU
2151 - Initialize Animals
2162 - Change Age
2173 - Change Value
2184 - Display
2190 - EXIT
220Enter Selection: 1
221
222Are you sure you want to reinitialize (Y/N)? x
223
224**** x is an invalid entry ****
225**** Please Enter Y or N ****
226
227Are you sure you want to reinitialize (Y/N)? n
228Animals have not been reinitialized!
229
230MENU
2311 - Initialize Animals
2322 - Change Age
2333 - Change Value
2344 - Display
2350 - EXIT
236Enter Selection: 4
237
238ANIMAL      NAME      AGE      VALUE
239-----
240Sheep      Fluffy      2      154154.52
241Sheep      Maa         4      165165.61
242Pig        Babe         3      123123.12
243
244MENU
2451 - Initialize Animals
2462 - Change Age
2473 - Change Value
2484 - Display
2490 - EXIT
250Enter Selection: 1
251
252Are you sure you want to reinitialize (Y/N)? Y
253Initializing Fluffy, Maa, Babe...
254
255MENU
2561 - Initialize Animals

```

output.txt

```
257 2 - Change Age
258 3 - Change Value
259 4 - Display
260 0 - EXIT
261 Enter Selection: 4
262
263 ANIMAL      NAME      AGE      VALUE
264 -----
265 Sheep      Fluffy      1      15000.00
266 Sheep      Maa        3      16520.35
267 Pig        Babe        2      10240.67
268
269 MENU
270 1 - Initialize Animals
271 2 - Change Age
272 3 - Change Value
273 4 - Display
274 0 - EXIT
275 Enter Selection: 0
```

ClassHeader.h

```

1 /*****
2  * AUTHOR      :Daniel Olaes & Faris Hijazi
3  * STUDENT ID  :1128957 & 1039438
4  * LAB #12     :Intro to OOP
5  * CLASS       :CS1B
6  * SECTION     :MW: 7:30pm
7  * DUE DATE    :4/25/19
8  *****/
9
10 #ifndef CLASSHEADER_H_
11 #define CLASSHEADER_H_
12
13 #include <iomanip>
14 #include <string>
15 using namespace std;
16
17 class Animal
18 {
19     public:
20         Animal();
21         ~Animal();
22
23         void SetInitialValues(string aName, string aType, int aAge,
24                               float aValue);
25         void ChangeAge(int aAge);
26         void ChangeValue(float aValue);
27
28         void Display() const;
29         string GetName() const;
30         string GetType() const;
31         int GetAge() const;
32         float GetValue() const;
33
34     private:
35         string name;
36         string type;
37         int age;
38         float value;
39
40 };
41
42 #endif /* CLASSHEADER_H_ */
43

```

MainHeader.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 inputted. 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
```

MainHeader.h

```

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

```


MainHeader.h

```
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
```

main.cpp

```

1 /*****
2  * AUTHOR      :Daniel Olaes & Faris Hijazi
3  * STUDENT ID  :1128957 & 1039438
4  * LAB #12     :Intro to OOP
5  * CLASS       :CS1B
6  * SECTION     :MW: 7:30pm
7  * DUE DATE    :4/25/19
8  *****/
9
10 #include "MainHeader.h"
11
12 /*****
13  * LAB 12 - INTRO TO OOP
14  * -----
15  *
16  *****/
17
18 int main()
19 {
20     /*****
21      * VARIABLES
22      *****/
23     Animal fluffy; //CALC - instance of animal class
24     Animal maa;    //CALC - instance of animal class
25     Animal babe;   //CALC - instance of animal class
26
27     int menuOption; //IN - menu option input by user
28     char initialCon; //IN - gets initial conditions for initialization
29     int animalOption; //IN - gets animal option user selected
30     int newAge; //IN - value of new age input by user
31     float newValue; //IN - new animal value input by user
32
33     PrintHeaderOStream ( cout, "Intro to OOP", 12, 'L');
34
35     menuOption = InputMenuOp('S');
36
37     if(menuOption == InitializeAnimals)
38     {
39         cout << "\nInitializing Fluffy, Maa, Babe...\n";
40         fluffy.SetInitialValues("Fluffy", "Sheep", 1, 15000.00);
41         maa.SetInitialValues("Maa", "Sheep", 3, 16520.35);
42         babe.SetInitialValues("Babe", "Pig", 2, 10240.67);
43
44         menuOption = InputMenuOp('M');
45
46         while(menuOption != 0)
47         {
48
49             switch(menuOption)
50             {
51                 case InitializeAnimals:initialCon = ConfirmReinitialize();
52                 if(initialCon == 'Y')
53                 {
54                     cout << "Initializing Fluffy, Maa, "
55                     << "Babe...\n";
56                     fluffy.SetInitialValues("Fluffy",
57                                             "Sheep",
58                                             1,
59                                             15000.00);
60                     maa.SetInitialValues("Maa",
61                                         "Sheep",
62                                         3,
63                                         16520.35);
64                     babe.SetInitialValues("Babe",

```

```

main.cpp

65         "Pig",
66         2,
67         10240.67);
68     }
69     else
70     {
71         cout << "Animals have not been "
72             << "reinitialized!\n";
73     }
74     break;
75 case AgeChange: animalOption = InputAnimalSelected('A');
76     newAge = InputNewAge();
77     cout << "\nChanging ";
78     switch(animalOption)
79     {
80         case AnimalOne: cout << "Fluffy's ";
81                         fluffy.ChangeAge(newAge);
82                         break;
83         case AnimalTwo: cout << "Maa's ";
84                         maa.ChangeAge(newAge);
85                         break;
86         case AnimalThree: cout << "Babe's ";
87                           babe.ChangeAge(newAge);
88                           break;
89     }
90     cout << "age to " << newAge << "... \n";
91     break;
92
93 case ValueChange: animalOption = InputAnimalSelected('V');
94     newValue = InputNewValue();
95     cout << "\nChanging ";
96     switch(animalOption)
97     {
98         case AnimalOne: cout << "Fluffy's ";
99                         fluffy.ChangeValue(newValue);
100                        break;
101         case AnimalTwo: cout << "Maa's ";
102                         maa.ChangeValue(newValue);
103                         break;
104         case AnimalThree: cout << "Babe's ";
105                           babe.ChangeValue(newValue);
106                           break;
107     }
108     cout << fixed << setprecision(2);
109     cout << "value to " << newValue << "... \n";
110     cout.unsetf(ios::fixed);
111     cout << setprecision(6);
112     break;
113
114 case DisplayAnimals: DisplayAnimalHeader();
115     fluffy.Display();
116     maa.Display();
117     babe.Display();
118     break;
119 case Exit: break;
120 }
121
122 menuOption = InputMenuOp('M');
123
124 }
125 }
126 }
127

```

ClassMethods.cpp

```

1  /*****
2  * AUTHOR      :Daniel Olaes & Faris Hijazi
3  * STUDENT ID  :1128957 & 1039438
4  * LAB #12     :Intro to OOP
5  * CLASS       :CS1B
6  * SECTION     :MW: 7:30pm
7  * DUE DATE    :4/25/19
8  *****/
9
10 #include "MainHeader.h"
11 #include "ClassHeader.h"
12
13 /*****
14 * CONSTRUCTOR Animal
15 * -----
16 *
17 * RETURN: NA
18 *****/
19 Animal::Animal()
20 {
21     name.clear();
22     type.clear();
23     age = 0;
24     value = 0;
25 }
26
27 //-----
28
29 /*****
30 * DECONSTRUCTOR Animal
31 * -----
32 *
33 * RETURN: NA
34 *****/
35 Animal::~Animal()
36 {
37     //NOTHING
38 }
39
40 //-----
41
42 /*****
43 * METHOD SetInitialValues
44 * -----
45 * sets the initial values for an object of class animal.
46 * RETURN: NA
47 *****/
48 void Animal::SetInitialValues(string aName,
49                                string aType,
50                                int aAge,
51                                float aValue)
52 {
53     name = aName;
54     type = aType;
55     age = aAge;
56     value = aValue;
57 }
58
59 //-----
60
61 /*****
62 * METHOD ChangeAge
63 * -----
64 * changes the age of an object of class animal.

```

```

65  * RETURN: NA
66  *****/
67 void Animal::ChangeAge(int aAge)
68 {
69     age = aAge;
70 }
71
72 //-----
73
74 /*****
75  * METHOD ChangeValue
76  *-----
77  * changes the value of an object of class animal.
78  * RETURN: NA
79  *****/
80 void Animal::ChangeValue(float aValue)
81 {
82     value = aValue;
83 }
84
85 //-----
86
87 /*****
88  * METHOD Display
89  *-----
90  * displays all atributes of an object of class animal
91  * RETURN: NA
92  *****/
93 void Animal::Display() const
94 {
95     /*****
96     * CONSTANT
97     *-----
98     * ANIMAL_COL - size of animal col
99     * NAME_COL   - size of name col
100    * AGE_COL     - size of age col
101    * VALUE_COL   - size of value col
102    *****/
103    const int ANIMAL_COL = 11;
104    const int NAME_COL   = 15;
105    const int AGE_COL     = 3;
106    const int VALUE_COL   = 12;
107
108    cout << left;
109    cout << setw(ANIMAL_COL) << type;
110    cout << setw(NAME_COL)   << name;
111    cout << right;
112    cout << setw(AGE_COL)    << age;
113    cout << fixed << setprecision(2);
114    cout << setw(VALUE_COL) << value << endl;
115    cout.unsetf(ios::fixed);
116    cout << setprecision(6);
117 }
118
119 //-----
120
121 /*****
122  * METHOD GetName
123  *-----
124  * returns name atribute.
125  * RETURN: name
126  *****/
127 string Animal::GetName() const
128 {

```

```

129     return name;
130 }
131
132 //-----
133
134 /*****
135  * METHOD GetType
136  *-----
137  * returns type attribute.
138  * RETURN: type
139  *****/
140 string Animal::GetType() const
141 {
142     return type;
143 }
144
145 //-----
146
147 /*****
148  * METHOD GetAge
149  *-----
150  * returns age attribute.
151  * RETURN: age
152  *****/
153 int Animal::GetAge() const
154 {
155     return age;
156 }
157
158 //-----
159
160 /*****
161  * METHOD GetValue
162  *-----
163  * returns value attribute.
164  * RETURN: value
165  *****/
166 float Animal::GetValue() const
167 {
168     return value;
169 }
170

```

ErrorCheckingFunctions.cpp

```

1  /*****
2  * AUTHOR      :Daniel Olaes & Faris Hijazi
3  * STUDENT ID  :1128957 & 1039438
4  * LAB #12     :Intro to OOP
5  * CLASS       :CS1B
6  * SECTION     :MW: 7:30pm
7  * DUE DATE    :4/25/19
8  *****/
9
10 #include "MainHeader.h"
11
12 /*****
13 * FUNCTION InputMenuOp
14 * -----
15 * This function allows the user to select a menu option and validate the menu
16 * option inputted. If menuType is 'S', then the small menu will be output and
17 * the user input validated according to the small menu. If menuType is 'M', the
18 * main menu will be output and the user input validated according to the main
19 * menu.
20 *
21 * RETURN: the validated menu options selected by the user
22 *****/
23
24 int InputMenuOp(char menuType) //IN - the type of menu option
25 {
26     /*****
27     * CONSTANTS
28     * -----
29     * ERROR_MESSAGE - the number of spaces reserved for the error message
30     *****/
31     const int ERROR_MESSAGE = 43;
32
33     /*****
34     * VARIABLES
35     *****/
36     int menuOp; //IN, CALC, OUT - the menu option selected by the user
37     string menuError; //CALC - the string that holds the error message
38     bool menuValid; //CALC - the boolean that checks the validity of
39                     // the list
40     if(menuType == 'S')
41     {
42         do
43         {
44             cout << "MENU\n"
45                  << "1 - Initialize Animals\n"
46                  << "0 - EXIT\n"
47                  << "Enter Selection: ";
48
49             menuValid = false;
50
51             cout << left;
52
53             if(!(cin >> menuOp))
54             {
55                 cout << setw(ERROR_MESSAGE)
56                      << "\n*** Please input a NUMBER between 0 and 1 ***"
57                      << endl << endl;
58                 cin.clear();
59
60                 cin.ignore(numeric_limits<streamsize>::max(), '\n');
61             }
62             else if(menuOp < 0 || menuOp > 1)
63             {
64                 menuError = "**** The number "

```

ErrorCheckingFunctions.cpp

```

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         else
124         {
125             menuValid = true;
126         }
127     }
128

```


ErrorCheckingFunctions.cpp

```

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();

```

ErrorCheckingFunctions.cpp

```

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 } while (!animalValid);
215
216 cout << right;
217
218 cin.ignore(1000, '\n');
219
220 return animalOp - 1;
221 }
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     else
282     {
283         ageValid = true;
284     }
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

```

ErrorCheckingFunctions.cpp

```

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     } while(!valueValid);
362
363     cout << right;
364
365     cin.ignore(1000, '\n');
366
367     return value;
368 }
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     /*****

```

ErrorCheckingFunctions.cpp

```

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

```

MiscFunctions.cpp

```

1  /*****
2  * AUTHOR      :Daniel Olaes & Faris Hijazi
3  * STUDENT ID  :1128957 & 1039438
4  * LAB #12     :Intro to OOP
5  * CLASS       :CS1B
6  * SECTION     :MW: 7:30pm
7  * DUE DATE    :4/25/19
8  *****/
9
10 #include "MainHeader.h"
11 #include "ClassHeader.h"
12
13 /*****
14 * FUNCTION PrintHeaderOStream
15 * -----
16 * This function receives an output type, assignment name, type and number then
17 * outputs appropriate header to the corresponding location - return nothing.
18 *
19 * RETURNS: nothing
20 * - the class header is output onto the screen IO.
21 *****/
22
23 void PrintHeaderOStream ( ostream &output, // IN - the type of output statement
24                          string asName,   // IN - assignment name
25                          int asNum,       // IN - assignment number
26                          char asType )    // IN - assignment type
27                          // ('L' = Lab
28                          // 'A' = Assignment)
29 {
30     output << left;
31     output << "*****\n";
32     output << " * PROGRAMMED BY : Daniel Olaes & Faris Hijazi\n";
33     output << " * " << setw(14) << "STUDENT ID" << ": 1128957 & 1039438\n";
34     output << " * " << setw(14) << "CLASS" << ": CS1B - MW - 7:30pm\n";
35     output << " * ";
36
37     if (toupper(asType) == 'L')
38     {
39         output << "LAB #" << setw(9);
40     }
41     else
42     {
43         output << "ASSIGNMENT #" << setw(2);
44     }
45
46     output << asNum << ": " << asName << endl;
47     output << "*****\n\n";
48     output << right;
49 }
50
51 //-----
52
53 /*****
54 * FUNCTION DisplayAnimalHeader
55 * -----
56 * This function will output the the animal list header to the screen IO.
57 *
58 * RETURN: nothing
59 * - the animal list header is printed to the screen IO
60 *****/
61 void DisplayAnimalHeader()
62 {
63     /*****
64     * 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

```