

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 //Added this workspace to allow us to use BasicTools and ValidationLibrary
7 using Week4_Class1;
8
9 namespace Week_6_Sample1_DataValidation
10 {
11     class Book
12     {
13         private string title;
14         private string authorFirst;
15         private string authorLast;
16         private string email;
17         private DateTime datePublished;
18         private int pages;
19         private double price;
20
21         protected string feedback;    //NEW - PROTECTED...CHILDREN SEE THIS BUT OTHERS DO NOT
22
23
24         public string Title
25         {
26             get
27             {
28                 return title;
29             }
30
31             set
32             {
33                 //Check for bad words...
34                 if (!ValidationLibrary.GotBadWords(value))
35                 {
36                     title = value; //If values does not have bad words...store it
37                 }
38                 else
39                 {
40                     feedback += "\nERROR: Title has a bad word in it."; //Else, leave Error Msg
41                 }
42             }
43         }
44
45
46         public string AuthorFirst
47         {
48             get
49             {
```

```
50         return authorFirst;
51     }
52
53     set
54     {
55         //Check for bad words...
56         if (!ValidationLibrary.GotBadWords(value))
57         {
58             authorFirst = value; //If values does not have bad words...store it
59         }
60         else
61         {
62             feedback += "\nERROR: Author's first name has a bad word in it."; //Else, leave Error Msg
63         }
64     }
65 }
66
67
68 public string AuthorLast
69 {
70     get
71     {
72         return authorLast;
73     }
74
75     set
76     {
77         authorLast = value;
78     }
79 }
80
81
82
83 public string Email
84 {
85     get
86     {
87         return email;
88     }
89
90     set
91     {
92         //Is the email format proper?
93         if (ValidationLibrary.IsValidEmail(value))
94         {
95             email = value; //Yes...store it
96         }
97         else
98         {
99             feedback += "\nERROR: Invalid email."; //Else...leave feedback
```

```

100         }
101     }
102 }
103
104
105 public DateTime DatePublished
106 {
107     get
108     {
109         return datePublished;
110     }
111
112     set
113     {
114         //If the date given is not a future date...
115         if (ValidationLibrary.IsAFutureDate(value) == false)
116         {
117             datePublished = value; //Past Date...store it
118         }
119         else
120         {
121             //Future Date...Store error msg in feedback
122             feedback += "\nERROR: You cannot enter future dates";
123         }
124     }
125 }
126
127
128
129 public int Pages
130 {
131     get
132     {
133         return pages;
134     }
135
136     set
137     {
138         //if we have the miimum number of pages needed...
139         if (ValidationLibrary.IsMinimumAmount(value, 0) == true)
140         {
141             pages = value; //store the # of pages
142         }
143         else
144         {
145             //Store an error msg in Feedback
146             feedback += "\nERROR: Sorry you entered an invalid # of
147                 pages.";
148         }
149     }
150 }

```

```
150
151
152
153
154     public double Price
155     {
156         get
157         {
158             return price;
159         }
160
161         set
162         {
163             if (ValidationLibrary.IsMinimumAmount(value, 1) == true)
164             {
165                 price = value;
166             }
167             else
168             {
169                 feedback += "\nERROR: Price is not sufficient.";
170             }
171         }
172     }
173
174     //NEW- Allows class to communicate with outside programs
175     public string Feedback
176     {
177         get { return feedback; }           //allows outside code to see feedback
178         // Notice there is no SET...This is because only the class can change ↗
179         feedback
180     }
181
182     //NEW - Default Constructor - Runs automatically when object instance ↗
183     // created
184     public Book()
185     {
186         //Initialize so that there are no nulls, especially feedback
187         title = "";
188         authorFirst = "";
189         authorLast = "";
190         pages = 0;
191         datePublished = DateTime.Parse("1/1/1500");
192         price = 0.0;
193         feedback = "";
194     }
195
196 }
197 }
198
```