```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//Added this workspace to allow us to use BasicTools and ValidationLibrary
using Week4_Class1;

namespace Week_6_Sample1_DataValidation
{
    class Book
    {
        private string title;
        private string authorFirst;
        private string authorLast;
        private string email;
        private DateTime datePublished;
        private int pages;
        private double price;

        protected string feedback;    //NEW - PROTECTED...CHILDREN SEE THIS BUT OTHERS DO NOT


        public string Title
        {
            get
            {
                return title;
            }

            set
            {
                //Check for bad words...
                if (!ValidationLibrary.GotBadWords(value))
                {
                    title = value;  //If values does not have bad words...store it
                }
                else
                {
                    feedback += "\nERROR: Title has a bad word in it.";  //Else, leave Error Msg
                }
            }
        }


        public string AuthorFirst
        {
            get
            {
                return authorFirst;
            }

            set
            {
                //Check for bad words...
                if (!ValidationLibrary.GotBadWords(value))
                {
                    authorFirst = value;  //If values does not have bad words...store it
                }
                else
                {
                    feedback += "\nERROR: Author's first name has a bad word in it.";  //Else, leave Error Msg
                }

            }
        }

        public string AuthorLast
        {
            get
            {
                return authorLast;
            }

            set
            {
                authorLast = value;
            }
        }



        public string Email
        {
            get
            {
                return email;
            }

            set
            {
                //Is the email format proper?
                if (ValidationLibrary.IsValidEmail(value))
                {
                    email = value;      //Yes...store it
                }
                else
                {
                    feedback += "\nERROR: Invalid email."; //Else...leve feedback
```

```
                      error msg
                }
            }
        }


        public DateTime DatePublished
        {
            get
            {
                return datePublished;
            }

            set
            {
                //If the date given is not a future date...
                if (ValidationLibrary.IsAFutureDate(value) == false)
                {
                    datePublished = value;  //Past Date...store it
                }
                else
                {
                    //Future Date...Store error msg in feedback
                    feedback += "\nERROR: You cannot enter future dates";
                }
            }
        }



        public int Pages
        {
            get
            {
                return pages;
            }

            set
            {
                //if we have the miimum number of pages needed...
                if (ValidationLibrary.IsMinimumAmount(value, 0) == true)
                {
                    pages = value;  //store the # of pages
                }
                else
                {
                    //Store an error msg in Feedback
                    feedback += "\nERROR: Sorry you entered an invalid # of
                     pages.";
                }
            }
        }
```

```csharp




        public double Price
        {
            get
            {
                return price;
            }

            set
            {
                if (ValidationLibrary.IsMinimumAmount(value, 1) == true)
                {
                    price = value;
                }
                else
                {
                    feedback += "\nERROR: Price is not sufficient.";
                }
            }
        }

        //NEW- Allows class to communicate with outside programs
        public string Feedback
        {
            get { return feedback; }        //allows outside code to see feedback
            // Notice there is no SET...This is because only the class can change feedback
        }


        //NEW - Default Constructor - Runs automatically when object instance created
        public Book()
        {
            //Initialize so that there are no nulls, especially feedback
            title = "";
            authorFirst = "";
            authorLast = "";
            pages = 0;
            datePublished = DateTime.Parse("1/1/1500");
            price = 0.0;
            feedback = "";
        }


    }
}

```