Contents

Solution

Simplest Solution

A possible solution is shared in this archive[1]:

```
using System;

class Bookmarker
{
  public string Title { get; set; }
  private int tPages; // Total number of pages.
  public int TPages // The property will check that the
   ↪  total number of page is "valid"
  {
    set
    {
      if (value <= 0) // This will throw an error if the
       ↪  value passed is negative.
        throw new ArgumentException(
          "The total number of pages cannot be negative."
        );
      else if (value < cPages) // This will throw an error
       ↪  if the value passed is less than the current
       ↪  page.
        throw new ArgumentException(
          "The total number of pages cannot be less than
           ↪  the current page."
        );
      else
        tPages = value; // If no errors were thrown, we
 ↪  set the value to the value passed.
    }
    get { return tPages; }
  }
  private int cPages; // Current page
  public int CPages // The property will check that the
   ↪  current number of page is "valid"
```

[1] https://princomp.github.io/code/projects/Bookmarker.zip

```
{
  set
  {
    if (value < 0)
      throw new ArgumentException(
        "You cannot have read a negative number of
        ↪  pages!"
      );
    else if (value > tPages)
    {
      throw new ArgumentException(
        "You cannot have read more than the total number
        ↪  of pages!"
      );
    }
    else
      cPages = value;
  }
  get { return cPages; }
}

public Bookmarker(string titleP, int tPagesP, int
 ↪  cPagesP)
{
  Title = titleP;
  TPages = tPagesP;
  CPages = cPagesP;
}

public override string ToString()
{
  return $"You have read {((double)cPages / tPages):P}
   ↪  of \"{Title}\".\nYou have {(1 - ((double)cPages /
   ↪  tPages)):P} to go!";
}

public void Read(int pReadP)
{
  if (pReadP + cPages > tPages)
    throw new ArgumentException(
      "You cannot have read more than the total number
       ↪  of pages!"
    );
  else
    cPages += pReadP;
}
```

```
}
using System;

class Program
{
  static void Main()
  {
    string title,
      tPages,
      cPages;
    /*
     * To trigger failure to create the object,
     * test with the following values:
     */
    // "Random", "Test", "0"    to get "Input string was
     ↪  not in a correct format."
    // "Random", "-12",  "0"    to get "The total number
     ↪  of pages cannot be negative."
    // "Random", "12",   "Test" to get "Input string was
     ↪  not in a correct format."
    // "Random", "12",   "15"   to get "You cannot have
     ↪  read more than the total number of pages!
    // "Random", "12",   "-12"  to get "You cannot have
     ↪  read a negative number of pages!"
    /*
     * To trigger error when calling the "Read" method,
     * test with the following values, after having
 ↪  created an object
     * using "Test", "10", "5"
     */
    // "6"    to get "You cannot have read more than the
     ↪  total number of pages!"
    // "-3"   to get "Input string was not in a correct
     ↪  format."
    // "Test" to get "Input string was not in a correct
     ↪  format."
    /*
     * An additional test would be to add, for example
     *             book1.TPages = "3";
     * after
     *             Console.WriteLine(book1);
     * to make sure that setting the number of page to an
 ↪  "invalid"
     * value would trigger the error
```

```
     * "The total number of pages cannot be less than the
↪  current page."
     */

    try
    {
      Console.WriteLine("Enter the title of the book.");
      title = Console.ReadLine();
      Console.WriteLine("Enter the total number of
↪  pages.");
      tPages = Console.ReadLine();
      Console.WriteLine(
        "Enter the page you stopped your reading at."
      );
      cPages = Console.ReadLine();
      Bookmarker book1 = new Bookmarker(
        title,
        int.Parse(tPages),
        int.Parse(cPages)
      );
      Console.WriteLine(book1);
      Console.WriteLine("How many pages did you read?");
      book1.Read(int.Parse(Console.ReadLine()));
      Console.WriteLine(book1);
    }
    catch (Exception ex)
    {
      Console.WriteLine(ex.Message);
    }
  }
}
```

(Download this code)[2]

A solution completing the bonus

> "Have your class handle strings, so that, for example, your Bookmarker constructor would take three strings as input"

is shared in this archive[3]:

[2] https:/princomp.github.io/code/projects/Bookmarker.zip
[3] https:/princomp.github.io/code/projects/Bookmarker_with_strings.zip