Larger Assignment, Part 1.

"A library implies an act of faith."
                                    – Victor Hugo

To begin with our library building project, we are going to make two classes this week: Author and Book.

The tester sample code is online. Do not modify it, simply create the two classes and the associated methods it expects to run.

An **Author** is a writer of books.
They have a first and last name and birth/death dates.

An **Author** has two constructors:

```
Author()
```
    blank constructor, creates a new Author with no name, birth, or death dates set
```
Author( last name, first name )
```
    creates a new Author with a name, but birth and death dates not set

An **Author** has at least the following methods:
(You can make more if you feel you need to, for helping your other methods, but I will be calling and testing these.)

```
int getBirth()
```
    returns an int representing the author's year of birth
```
int getDeath()
```
    returns an int representing the author's year of death
```
void setDates( birth )
```
    sets the author's year of birth, must be a valid year
```
void setDates( birth, death )
```
    sets both the author's year of birth and death, must be valid years, and the year of death must be after the year of birth

```
int compareTo( Author )
```
compares two authors. This is similar to Java's String compareTo() method. If two authors have the same first name and last names it should return 0, otherwise it should return not 0.

Bonus: Sometimes authors do not always write out their entire first name. For example Julian Barnes and J Barnes could be considered equivalent. However, Julian Barnes and John Barnes should not be. Make your compareTo more flexible by returning 0 if a first initial and last name match.

```
String toString()
```
Should return a string the author, of the form: Barnes, Julian

```
String printAuthorInfo()
```
Should **return** (not print) a string the author, of the form:
```
Heaney, Seamus
```
If the year of birth (but not death) is set it should return:
```
Heaney, Seamus (b. 1939)
```
If both the years of birth and death have been set, it should return:
```
Heaney, Seamus (1939-2013)
```

Note a valid year, is one after the year -2000 (I am letting that represent BC) and before the year 2018 (some books do announce their publication in advance, though rarely more than five years out, most people have not forecast their births or deaths more than five years out).

A Book is our main object.
Books in our system have a date (year) of being written, a title, an ISBN, and a single Author (Author object) who wrote them.

A **Book** has three constructors:

```
Book()
```
blank constructor, creates a new Book with no title, year of publication, Author, or ISBN set
```
Book( title )
```
creates a new Book with a title, but no year of publication, Author, or ISBN set
```
Book( title, Author )
```
creates a new Book with a title and an Author (as an Author object) but no year of publication or ISBN set

A **Book** has at least the following methods:
(Again, you can make more if you feel you need to, for helping your other methods, but I will be calling and testing these.)

```
void setTitle( title )
```
     sets the book's title must be a not-empty title
```
String getTitle()
```
     returns a String representing the book's title
```
void setAuthor( Author )
```
     sets the Author object
```
Author getAuthor()
```
     returns the Author object who wrote the book
```
void setYear( year )
```
     sets the year the book was written/published, must be a valid year
```
int getYear()
```
     returns the year the book was written
```
void setISBN ( ISBN )
```
     sets the ISBN for the book, this must be of length 13 or length 10 (we will be omitting all dashes)
```
String getISBN()
```
     returns a String representing the ISBN of the book
```
boolean sameAuthor( Book )
```
     returns a boolean which is true if this book has the same Author as the book that was passed in in the arguments, or otherwise false (hint: you wrote a method to compare authors.)
```
int compareTo ( Book )
```
     compares two Book objects, for now this can be done ONLY by comparing ISBNs.
```
String toString()
```
     Should return (not print!) a String representation of this Book.
     If only a title has been set, return a string of the form:
          `The Arcades Project.`
     If a title and author are set:
          `The Arcades Project. Benjamin, Walter.`
     If a title and author and year have been set:
          `The Arcades Project (2002). Benjamin, Walter.`

     Note the punctuation above, it must be exact to pass the tests.

Note, when trying to run a setter method with an unallowable variable, please keep the previous value. Returning error messages is a good idea, however the text of them is up to you.

Finally, there are a series of constants at the top of the sample file. Please use these variables for properties of books and authors that have not yet been set.

Final output if achieving all points:


```
Attempting constructors:
  - Author constructors seem functional:  3/3
  - Book constructors seem functional:    3/3

Attempting simple getters/setters:
  - getTitle/setTitle:                    2/2
  - getAuthor/setAuthor:                  2/2
  - getYear/setYear:                      2/2
  - getISBN/setISBN:                      2/2
  - author getBirth/getDeath/setDates:  4/4

Attempting un-set values register correctly:
  - un-set values:  5/5

Attempting bad-value setters:
  - year too far in the future rejected:  2/2
  - empty title rejected:                 2/2
  - ISBN of incorrect length rejected:    2/2
  - year of birth in super past rejected: 2/2
  - year of death before birth rejected:  2/2

Comparisons:
  - author comparison tests:          3/3
  - *bonus* author comparison tests:  5/5
  - sameAuthor (across two books):    2/2
  - comparing two books:              2/2

Printing authors and books:
  - printing an author's name:          2/2
  - printing an author's name + info:  3/3
  - printing a book's name + info:      5/5

Overall Score: 55/50
   (can be up to 55 with bonus)
```