

## Homework Assignment 4

Your assignment for Homework 4 is to use a stack to parse an input XML file, check the XML for validity, and to use the data to create data payload instances from the input XML.

The driver program is given in the code on the website. You must implement a `ParseTheData` class that includes the methods invoked by the driver.

### Parsing XML

In Lab 02 you parsed XML data, so you can use that code as part of the code to read and parse XML and to separate tags from real data.

### The Data

The data for this assignment is the “N” chapter from Dr. Ward Briggs’s book *Biographical Dictionary of North American Classicists*. The original book has been reformatted (partly) as XML in order to put the book on the web as searchable content. You do not need to know much about what the data content actually is; what is important is that you can parse the data for correct nesting and that you can extract the actual data to place into fields.

### Processing

- Read the XML line by line.
- Determine if the incoming token is an XML tag or real data.
- If the token is an open tag, push it onto the stack.
- If the token is just data, push it onto the stack.
- If the token is a close tag, then pop the stack until you get the first open token and compare the token against the close to ensure that they match.
- Before you start reading the actual data, you should first have read the baby version of a Document Type Definition, which in this case is simply a list of the XML tags declared to be legal for this application.

- As you are reading tokens and popping the stack: after having determined that with a closing tag you have XML that is properly nested, you should add the data into “the current instance” of **Biography** and set instance variables accordingly in the instance of **Biography**. If you finish off a person with the closing tag for all the data for that person, then you should add that instance of **Biography** to an **ArrayList** and create a new instance of **Biography** for the next set of reads.

## **XMLData**

You are required to implement the **IXMLData** interface in your **XMLData** class.