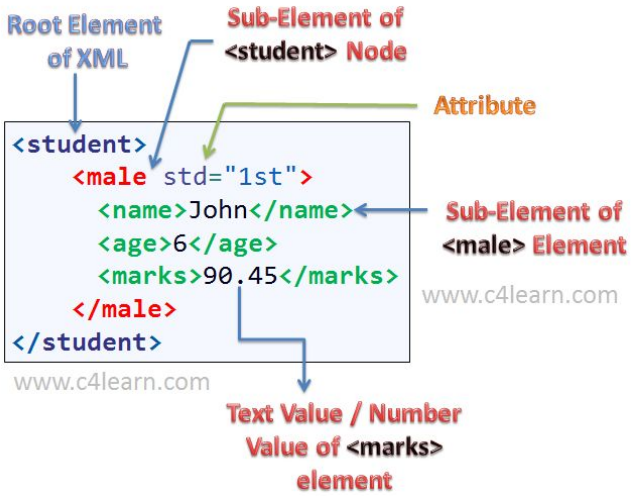
CSL202 | Lab exam 3 | 12/Apr/2018 | 100 points

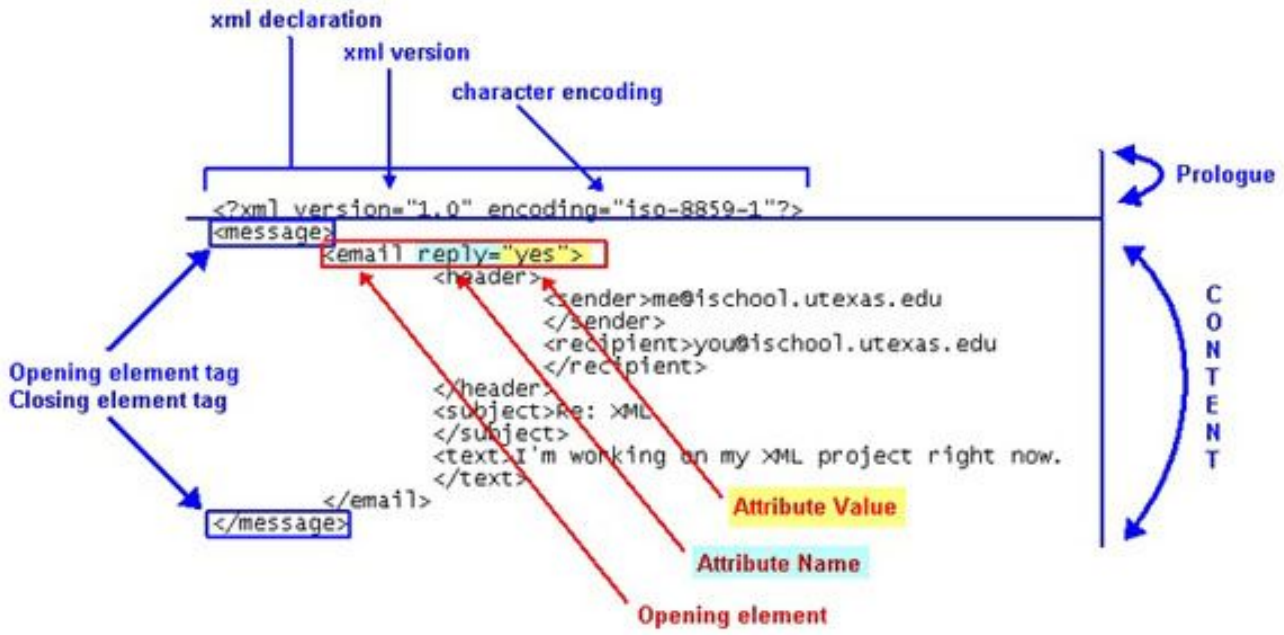
- Important instructions for coding submission are here: https://goo.gl/IMWvdF
- **Late submissions will not be considered.**
- **DO NOT copy code from online tutorials etc. However, you may refer to the javadocs at https://docs.oracle.com/javase/7/docs/api/index.html**
- Your submission should be packaged as a zip file named **exactly** in this format: CSL202-[your entry no.]-[assignment no.].zip.

We need to write a program in Python which can check *well formedness* of an XML file which will be supplied as input to the program via command line argument. You can assume the following definition of *well formedness*:

1. The begin, end, and empty-element tags that delimit the elements are correctly nested, with none missing and none overlapping (see examples below).
2. The element tags are case-sensitive; the beginning and end tags must match exactly. Tag names contain only a-z or A-Z.
3. There is a single "root" element that contains all the other elements.

Structure of a typical XML document:





Examples of well formed and non well formed XML:

1. *<!-- WRONG! NOT WELL-FORMED XML! -->*
   ** **This one ****has ****overlapping**** strong tag with em** **
2. ** **This one ****has ****unclosed strong tag** **
3. **<P>**This one **<em12>**has different case for opening and closing P, and em12 has digits in the tag name**</em12> **
4. *<!-- Correct: Well-formed XML. -->*
   ** **This is ****proper ****XML**** ****document** **
5. ** **Alternatively ****emphasized**** ****strong emphasized**** strong** **

Your program should print “well formed” or “Not well-formed” on screen based on how it decides about the well formedness of the input XML file.

NOTE: You should <u>not</u> use any XML or markup language processing libraries for implementing your solution. You can make use of regular expressions if you want.