**Parsing and Manipulating XML Data**

XML is the basis for many interfaces and web services. Consequently, reading and manipulating XML data is a common task in software development.

Description

An online plant distributor has recently experience a shortage in its supply of Anemone plants such that the price has increased by 20%. Their plant catalog is maintained in an XML file and they need a Python utility to find the plant by name, read the current price, change it by the specified percentage, and update the file. Writing this utility is your assignment.

Using Python’s ElementTree XML API, write a Python program to perform the following tasks below. Note that your program’s execution syntax must be as follows:

python xmlparse.py plant\_catalog.xml plantName percentChange

1. Using ElementTree, read in this assignments XML file plant\_catalog.xml specified by a command line parameter as shown above.
2. Find the plant by the name passed in as an argument on the command line (plantName above).
3. Once found, read the current price and adjust it by the command line argument percentChange. Note that this value could be anything in the range of -90 < percentChange < 100.

For example, if you run your script as follows:

python plant\_catalog.xml "Greek Valerian" -20

with the original XML containing:

<PLANT>

<COMMON>Greek Valerian</COMMON>

<BOTANICAL>Polemonium caeruleum</BOTANICAL>

<ZONE>Annual</ZONE>

<LIGHT>Shade</LIGHT>

<PRICE>4.36</PRICE>

<AVAILABILITY>071499</AVAILABILITY>

</PLANT>

The resulting file should contain:

<PLANT>

<COMMON>Greek Valerian</COMMON>

<BOTANICAL>Polemonium caeruleum</BOTANICAL>

<ZONE>Annual</ZONE>

<LIGHT>Shade</LIGHT>

<PRICE>3.48</PRICE>

<AVAILABILITY>071499</AVAILABILITY>

</PLANT>

**Note:** You may reduce the precision of the calculation if you wish but it isn’t required.

**Hints**

Since XML is just a text file, you could write the code to read all the data and the decode the XML information. However, I certainly don’t recommend this approach. Instead, let Python do it for you! Using Python’s ElementTree module, parse the file into an “in-memory” representation of the XML data. Once parsed, the root (or starting place) is as simple as requesting it from the tree. Once you have the root, you can call methods to find what you are looking for and modify them appropriately. You'll want to "findall" the plants and, "for" each plant "in" the result, you'll want to compare the name with the name passed on the command line. If you find a match you'll apply the percentage change, save the result back to the tree.

When you are done with the search you will "write" the tree back to a file. I suggest using a different file name or you will be having to re-download the original with each run.

One note of caution, be sure to read about XML in the Distributed Systems text. From doing so and reviewing the data file you will not that there are no attributes in the XML file. Consequently, you do not need to use attribute methods when you attempt this assignment.

The following code snippet will give you a good starting point:

# Calling arguments: plant\_catalog.xml plantName percentChange

import xml.etree.ElementTree as ET

import sys

# input parameters

searchName = sys.argv[2]

percent = float(sys.argv[3])

# parse XML data file

tree = ET.parse(sys.argv[1])

root = tree.getroot()

Submit your Python source code in by 11:59 p.m. (ET) on Monday of Module/Week 5.