CS 2316
Individual Homework 10 – XML Processing
Due: Monday April 16th,   before 11:55 PM
Out of 100 points

Files to submit: **1. HW10.py**

This is an INDIVIDUAL assignment!
Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 2316, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:
-            -   TA Helpdesk – Schedule posted on class website.
-            -   Email TA's or use T-Square

ForumsNotes:
- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- **Do not wait until the last minute** to do this assignment in case you run into problems.

Background:
You are hired by a consulting firm which specializes in providing support for obsolete versions of enterprise software that is no longer supported by the original manufacturer. Companies pay your employer for support contracts so that they do not have to spend even more money upgrading to the latest version of the software. In some cases, the original manufacturer has discontinued support for a particular piece of software entirely, leaving your employer as the only support option for businesses who still use the old software.

Every Wednesday your job is to visit the IBM website and make a spreadsheet for your boss that lists all of the software that will no longer be supported within the next year. You have found a website that offers you this information in XML format. If you could just download the process the information automatically using Python, you would be able to play Minecraft most of the rest of the day.

The webpage is: http://www-01.ibm.com/software/support/lifecycleapp/xmldownload.jsp
On this webpage you can select a certain number of days of updates or use the following
URL to download a complete list of data (updated weekly):
http://www.ibm.com/software/support/lifecycle/rss/PLCWeeklyXMLDownload.xml

You will also find the link to the XML schema which is useful for understanding what
the data means: http://www-
01.ibm.com/software/support/lifecycleapp/IbmPlcDownload_v2.xsd

Data Download

Your program should download the complete product lifecycle XML file from the URL
above. If you are unable to download the data (due to network problems), you should
catch any exceptions and show the user a warning dialog.

Data Filtering

If you are successful in downloading the data, your program should examine all products.
**You are looking for any versions that will no longer be supported by IBM within
365 days from the current date.** (The End of Service date is today or later, but less than
or equal to 365 days in the future). Any software titles and versions that fall in this
category should be exported as explained below. NOTE: Your boss is not interested in
software that has already fallen out of service, so do not bother reporting versions where
the EOS date is before the current date!

Note that it is possible that a software product will have multiple versions. Each version
is then broken down into multiple releases, and each release can have different "Mod
Level Numbers". For example, the "AIX Fast Connect for POWER" software has two
release mods under version 3, specifically 3.2.x and 3.1.x. Version 3.1.x has an EOS date
of 30 Sep 2008, while version 3.2.x does not have an EOS date. Neither of these two
versions should appear in your spreadsheet. (Versions 3.2.x because the EOS **date is
blank**, and 3.1.x because the **EOS date is in the past**, not within 1 year of the current
date!)

An example that SHOULD appear in the output data is "Tivoli SANergy", version 2,
release 2, mod level x, which has an EOS data of '30 Apr 2013'.

Date Math

Python includes libraries that will make it easier to to "date math" to find out if a certain
date is within 365 days (in the future) of the current date. We suggest you look at the
datetime module, and specifically the *date* and *timedelta* objects. Note that timedelta
objects have a *days* class variable that will be useful to you. You will have to convert
IBM's date format (DD MON YYYY) into a date object before you can add or subtract it

from the current date. Note that IBM uses 3 letter month abbreviations, while the date object expects a number. You may want to use a dictionary to convert between three letter abbreviations and the appropriate numbers.

Output

After your program finds all software products (and versions) that have an End of Service (EOS) date between the current date and one year from the current date, it should produce an XML file containing the data. See the description of the WriteXML function below for the exact output format specification.

GUI

Your program must have a GUI with a "Download File" button. When you press the "Download File" button, your program should download the complete product lifecycle XML file from the IBM website. If it fails to download due to a network error, a warning dialog should pop up, so the user can reconnect to the Internet. Otherwise, the following window should pop up.
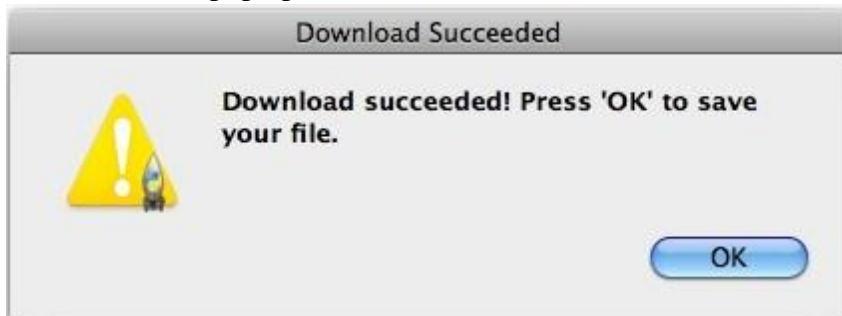
Pressing the 'OK' button will open up another GUI window which looks like the following. If the user presses 'Yes', a *file save dialog* will open and ask the user where to save the output file. Otherwise, it will go back to your first GUI window with the "Download File" button.

The user should be able to enter the name of a file (such as "filteredIBM.xml") and your program will save the results as an XML file. After the save is successful, you should update the GUI to say "File successfully saved!".

How to organize your code:

You may organize your code in any way you would like. We suggest you separate the problem up into the following functions. Writing the solution as multiple functions will allow you to break the big problem down into smaller, more manageable steps. You can test each function independently to make sure it works before moving on to the next one. **init** – An initializer function that sets up your GUI.

clicked - When the user clicks your "Download File" button, it should start the download, filter, and save functionality, and then update the GUI to report success.

download – A function that downloads the IBM XML file and parses the XML to find the Software Title, Version Number, Release Number, ModLevelNumber and End of Service date (if any). It will store all of this data into a list, and pass it to the filter function.

parseIBMDate – A function that takes an IBM end of service string (in the format: "15 Apr 2009") and returns a python date object that represents that date.

filter – A function that processes each record in a list to determine if it matches our criteria or not. It can call the parseIBMdate function to convert the EOS text for each record into a date, and then compare it to the current date to find out if it is within one year in the future (IBMDate – currentDate should be between 0 and 365 days...negative days means the EOS is in the past, and days larger than 365 means it is more than 1 year in the future) Any records that match your criteria can be saved to a new list and returned.

writeXML – This function takes in a list of records that has been filtered (where each record consists of a Swtitle, Version Number, Release Number, ModLevel Number and EOS text), and writes those records out to an XML file. The root software should be named "Packages" and it will contain multiple records (e.g. "Package") where its text is the software title and its attributes are eosDate, modLevelNum, releaseNum, and versionNum. The first two records contained under "Packages" should look like the following.

```xml
- <Packages>
  - <Package eosDate="30 Sep 2013" modLevelNum="x" releaseNum="1" versionNum="6">
      Adapter for JD Edwards EnterpriseOne on WebSphere Application Server
    </Package>
  - <Package eosDate="30 Sep 2013" modLevelNum="x" releaseNum="1" versionNum="6">
      Adapter for Oracle E-Business Suite on WebSphere Application Server
    </Package>
```

Grading
You will earn points as follows for each piece of functionality in your code:

GUI   **15**

-Allows specification of Destination file location   5

-Shows all warning messages correctly and   10
 provides report to the user when done

Reading Data Source   **25**
-Downloads data correctly from website   20

-Proper Error Handling   5

XML Parsing / Filtering   **45**
-Software Titles are found   10

-Each version of each title is identified   10

-EOS dates are found for each version of the software   10

-Only titles/versions with an EOS date within 1 year are saved   15

Data Output   **15**
-Saves data correctly in XML format   15

Total Possible: **100**