CSE 445 Project4 (Assignments 7 & 8) (50+50 Points)

Fall 2018

Assignment 7 Questions 1, 2, 3, and 4 Due: Saturday, October 27, 2018, by 11:59pm
Assignment 8 Questions 5, 6, and 7 Due: Saturday, November 3, 2018 by 11:59pm

Plus a one-day grace period for each assignment.

Introduction

The aim of this project is to make sure that you understand and are familiar with the concepts covered in the lectures, including XML elements, attributes, statements, XML schema, XML validation, XML transformation (XSL), and the related library classes. By the end of the project, you should have applied these concepts and techniques in creating an XML file, its schema, its style sheet, and have written Web services and an SOA application to process these files.

This is an **individual project**. Each student must complete and submit independent work. No cooperation is allowed, even among the team members for project 3. Do not use WebStrar, which is shared among your team members to host your files, applications, or services in project 4. You can use your ASU personal Web site space to host your XML, XSD, and XSL files (see Part 0 question 4) and use localhost to host the Web applications and services.

Practice Exercises (No submission required)

No submission is required for this part of exercises. However, doing these exercises can help you better understand the concepts and thus help you in quizzes or exams.

1. Reading: Textbook Chapter 4.

2. Answer the multiple choice questions 1.1 through 1.16 of the Text Section 4.8. Study the material covered in these questions can help you to prepare for the class exercises, quizzes, and the exam.

3. Study for the questions 2 through 8 in Text Section 4.8. Make sure that you understand these questions and can briefly answer these questions. Studying the material covered in these questions can help you to prepare for the exam and understand the assignments.

4. If you have not activated your file service and personal Web hosting site at ASU, you can activate them at: www.asu.edu/selfsub. Choose "Subscribe to additional computing access/services", and then chose "Personal Webpage Hosting". To access your personal Web site, you need to have SSH or PuTTY software to connect to general.asu.edu. You can also search for "Uploading Your Personal Web page" within the ASU search page to find the steps for uploading your files. Notice that the ASU Personal Web site space hosts files only. You can use it to host your .xml, .xsd, and .xsl files. It does not host programs such as Web services and Web applications. IIS are not installed. In order for your files to be accessible from the Web, you must put the files into the WWW directory, it its sub-directories.

5 You can access the sample XML and XSD files at:

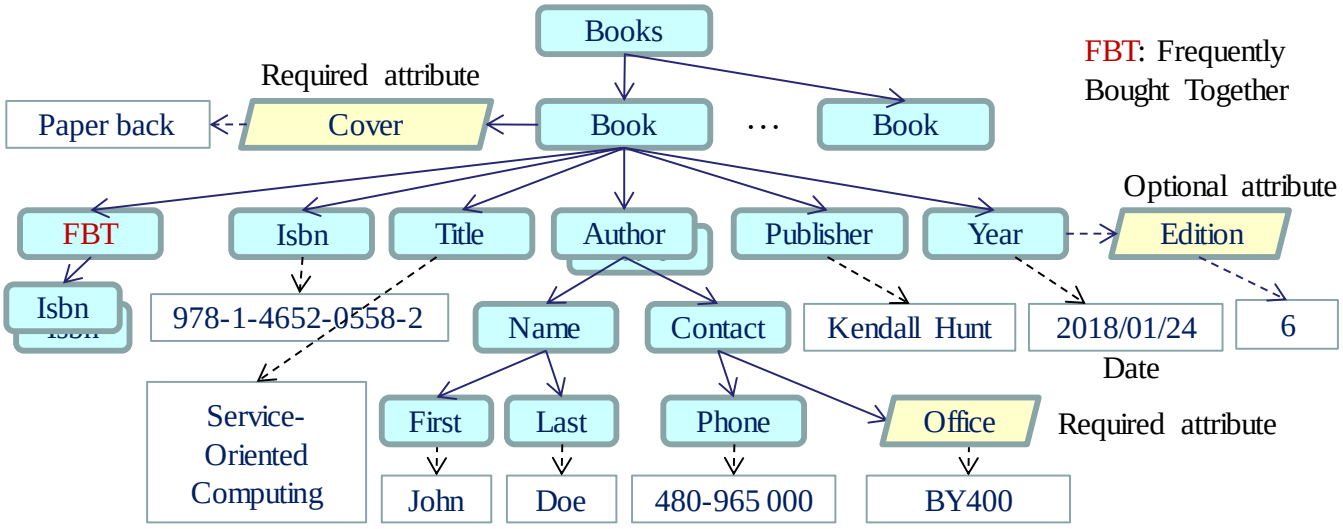
http://neptune.fulton.ad.asu.edu/WSRepository/xml/Courses.xml
http://neptune.fulton.ad.asu.edu/WSRepository/xml/Courses.xsd

Assignment 7: Creating XML, XSD, and XSL Files (50 points)

Questions 1, 2, 3, and 4 Due Date: Saturday, October 27, 2018, by 11:59pm (Arizona Time)

Assignment 7 has 50 points. You will create three files and an application. You will use the three files as the input to the programs in your assignment 8, and thus you must include them in your assignment 8 submission. You may modify these files when you resubmit them in assignment 8. However, we will not grade these files in assignment 8, which have been graded in assignment 7.

In this assignment, you will create a directory of Books, which can be represented as an XML tree. The diagram below shows the required structure of the directory of Books in XML tree notation that you will create in this assignment. All the "Book" elements have the same structure. Notice that different shapes and colors of boxes have different meanings. They represent elements, attributes, and text contents/values, respectively. The structure of elements and attributes given in the diagram below must be implemented as described, while the given text contents/values in the diagram are example's instance values and may be different in your files. The solid arrows show parent-child element relations, and the dotted arrows show the element-content relations. The optional attribute "Edition" means that the XML instance can have the attribute or not have the attribute, without causing a verification error against its schema. However, it is still required to define the optional attribute in the XML Scheme file. In the instances, you must provide the attribute for some books, but not for all books. For each Book element, you must allow multiple Author elements. The FBT (Frequently Bought Together books) element stores a list of Isbns from other books. These books must exist in the Book.xml.



1. Write the Books.xsd file that defines the XML schema allowing the structure shown in the diagram above. You can use any tool to create/edit the file. [10 points]

2. Create an XML file Books.xml as an instance of your schema file. Enter the information of at least twenty (20) real books into the Books.xml file. You can use any tool to edit the file. You must include all the elements given in the XML tree. If an element has a Required Attribute, you must provide the attribute for each of the elements. If an element has an Implied (Optional) Attribute, you will provide this attribute for some elements, but not for all elements. [10 points]

3. Write a Books.xsl file that defines the HTML style to display the books stored in Books.xml in a formatted table. You can choose your own format to present the data in the table, which must include all elements, attributes, and their values. You can use any tool to edit the XSL file. [10 points]

4. As a test page (TryIt page), create an ASPX Web site application (not a Web service) which takes the URL of an XML file as input (via textbox) and displays the element (tag) names, text contents, attribute names, and attribute values in the GUI page. You can define the order of data to be displayed. Make your Books.xml as the default input. [20 points]

Assignment 7 Blackboard Submission list: The three files Books.xml, Books.xsd, Books.xsl, and the complete solution folder on localhost (for Question 4) with all files that can be tested by the TAs.

In addition to the blackboard submission, the XML file needs to be deployed into a file server, so that it can be tested by your ASPX application. However, you still need to zip these three files and the Web site application into a zip file for Blackboard submission. Note: Do not use WebStrar server in this project. You can use your personal website offered by ASU at: http://www.public.asu.edu/~YourRITEID/ to host the files, see the Question 4 in Practice Exercises.

Assignment 8: Creating Web Services to Process XML, XSD, and XSL Files (50 points)

Assignment 8 Questions 5, 6, and 7 Due: Saturday, November 3, 2018 by 11:59pm

5. Develop a Web service (.svc) with **two** of the Web operations listed below. The node mentioned in the sub questions below includes every component (element, content, and attribute) shown in the XML tree in assignment 7. You need to choose two operations to implement. If you implement more than two operations, we will grade the first two operations only. If you have implemented and submitted a service listed below in your project 3 as an elective service, you cannot choose that service here.

5.1 Web operation "verification" takes the URL of an XML (.xml) file and the URL of the corresponding XMLS (.xsd) file as input and validates the XML file against the corresponding XMLS (XSD) file. The Web method returns "No Error" or an error message showing the available information at the error point. You must use files that you created in assignment 7, with and without fault injection, as the test cases. However, your service operation should work for other test cases too. [10 points]

5.2 Web operation "transformation" takes the URL of an XML (.xml) file and the URL of the XSL file as inputs and generates an HTML file or a string based on the XML and XSL files. The generated HTML file or string should be returned as the return value of the operation. In the TryIt GUI

(Question 6), you can store the returned file in a text file or in an .htm (or .html) file. You can also display the file in the GUI of your Web application as a table. You must use files that you created in assignment 7 as a test case. However, your service operation should work for other test cases (other XML and XSD files) too. [10 points]

5.3 Web operation "search" takes the URL of an XML (.xml) file and a key (e.g., the name of a book) as input. It returns the node's content information related to the search key, for example: name, contact, phone, etc. Your program must also read any attributes. Attributes should be searched too. If there are multiple occurrences, you must return all them. In this question, you can use DOM or SAX model. In the GUI (Question 6), you can display them all or once at a time through a "Display Next" button. [10 points]

5.4 Web operation "XPathSearch" takes the URL of an XML (.xml) file and a path expression as input. It returns the path expression value of the given path. It could be a list of nodes, the content value, etc., depending on the path given. [10 points]

5.5 Web operation "addBook" modifies the XML file by adding a new book into the XML file. You can use multiple parameters or one parameter that contains all the information required for adding a new book into the tree. The method must work for any tree that conforms with your XML schema file. After the modification, the tree must be saved back to the file in App_Data in your working directory. In the TryIt page, the added contents must be demonstrated. [10 points]

5.6 Web operation "FBT" returns the list of {book and title} of all the books in the FBT list of a given book Isbn. In the TryIt page in Question 6, you must provide a textbox for inputting and Isbn and an output space for the list of {book and title} of all the books in the FBT list. You need to search the book title based on Isbn. [10 points]

Notice that, for all the questions above, do not place the XSD file as a namespace in your XML file. It may cause an exception in some library classes. The absence of the XSD namespace will not cause a problem with the schema validation and with XSL transformation, as your method will take XML file as a parameter if needed.

6. Create a Web site application (ASPX), and add the project into the same "solution" that hosts your web services. The Web site application must provide a GUI (TryIt Page) which allows entering the required inputs, such as URLs and keyword, path, or contents, based on the questions that you select in Question 5. The GUI must have the buttons required to invoke the service operations, depending on the methods that you implement in the previous question, for example,

- The button that validates the XML file against the schema file;
- The button that generates the HTML file;
- The button that searches by keyword or by path in the XML file.
- The button that adds a new book.

The Web application must use the Web service created in Question 5 to perform the required processing, and display the return message in the GUI. You can use a textbox, a list box, a label, etc., to display the html file, or display it in a separate page. This assignment will be implemented on localhost / IIS Express. You must make ensure that the application and the services are still linked when the assignment is graded on a different computer. Do not deploy into the WebStrar. [20 points]

7. Testing: Based on your selection of the sub questions in question 5, provide your test inputs and test results. For example, if you chose questions 5.1, 5.2, or 5.3, you can place the files: Books.xml, Books.xsd, or Books.xsl, into a Web site and use them to test your program on your .Net development server. Inject an error into your XML file and make sure the validation service can detect the error. For question 7, you must submit the test results (inputs and outputs) in screenshots. For example: [10 points]

   (1) A screenshot of the GUI, including the output of the XML validation displayed in the GUI, with no error and with error messages;

   (2) A screenshot of the GUI, including the input and output for keyword search;

   (3) The Books.htm file (or Books.html) file generated.

   These three files above are not programs and can be deployed to any web location, such as the personal web space provided by ASU. See exercise 4 in Part 0. Do not use the server for assignment 3 in this assignment.

Blackboard Submission list for Part 2: The complete solution folder with all project files for the services and the application that can be tested by the TAs. You also need to submit the files used in the services, such as Books.xml, Books.xsd, Books.xsl, the files generated by the services, such as Books.htm (or Books.html). Put the screenshot of the testing results in a Word or PDF file. Zip all these files into a zip file for submission.

Submission Requirement

All submissions must be electronically submitted to the assignment folder where you downloaded the assignment paper. All files must be zipped into a single file.

Grading

We will grade each program or service following these steps:

(1) We will read the code and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

(2) Compile the code. If it does not compile, 40% of the points given in (1) will be deducted. For example, if you are given 20 points in step (1), your points will become 12 if the program fails to compile.

(3) If the code passes the compilation, we will execute and test the code. This can be on local machine or on the Web, depends on the assignment specification. If, for any reason, the program gives an incorrect output or crashes for any input, 20% of the points given in (1) will be deducted.

Please notice that we will not debug your program to figure out how big or how small the error is. You may lose 40% or 20% of your points for a small error such as missing a comma or a space!

Late submission deduction policy:

- No penalty for late submissions that are received within 24 hours of the given deadline;
- 1% grade deduction for every hour after the first 24 hours of the grace period!
- No submission will be allowed after Tuesday midnight.

Blackboard submission notice: The assignment consists of multiple distributed projects and components. They may be stored in different locations on your computer when you create them. You must copy these projects into a single folder for blackboard submission. To make sure that you have all the files included in the zip file and they work together, you must test them before submission. You must also download your own submission from the blackboard. Unzip the file on a different location or machine, and test your assignment and see if you can run the solution in a different location, because the TA will test your application on a different machine.