3460:209 Assignment 8-A

# **Assignment8-A: XML (eXtensible Markup Language)**

The purpose of this assignment is to help gauge your skills in writing small programs that involve vectors and c-string arrays. The program also contains functions and may perform input, output, files and file processing, flow of control, and/or calculations.

**PROGRAM SPECIFICATION**

The XML (eXtensible Markup Language) is a common format used to structure and store data on the web. The following is a small sample XML file that could be used to store names in an address book:

**<?xml version="1.0"?>**

**<address\_book>**

**<contact>**

**<name>George Clooney</name>**

**<street>1042 El Camino Real</street>**

**<city>Beverly Hills</city>**

**<state>CA</state>**

**<zip>90214</zip>**

**</contact>**

**<contact>**

**<name>Cathy Pearl</name>**

**<street>405 A St.</street>**

**<city>Palmdale</city>**

**<state>CA</state>**

**<zip>93352</zip>**

**</contact>**

For this program, you will download the ASSGN8-B.xml file. This file contains the XML we will use for our program.

The sample file contains many contacts. The <> tag denotes the start of a field and the </> tag denotes the end of the field.

a) You are hosting a party in Palmdale, CA. Write a program that reads in the address.xml file and outputs the names and addresses of everyone in Palmdale. Your program shouldn’t output any of the tag information, just the address content.

b) You would like to send an advertising flyer to everyone in zip codes 90210 through 90214. Write a program that reads in the address.xml file and outputs the names and addresses of everyone whose zip code falls within the specified range.

You may assume that each contact in the address file has the same structure and the same fields. However, your solution should be able to handle an input file with any number of contacts and should not assume that the fields within each contact are in the same order.

Make sure that your programs follow good documentation standards and follow the requirements for assignments. Reference the rubric standards on Brightspace. Note functions and data validation are now required. Do not use using namespace std;.

Submission Instructions – for programming solutions

On Brightspace, go to the matching Assignments for the ASSGN@-#, where @ is the chapter and # is the number or character of the problem assigned (eg., 5-11 for chapter 5, problem 11), and submit the program (cpp).

*Last updated 5.22.2016 by Will Crissey.*

*Be aware that programming falls under all of the rules of plagiarism. Be careful when using any coding found in the outside world that is not your own. Any evidence of plagiarism is subject to sanctions like forfeits, suspension, and even ejection, as determined by the Department of Student Conduct and Community Standards.*

**COPYRIGHT © 2017 Will Crissey, Jr., Assistant Professor, All rights reserved.**

This document is intended exclusively for use by The University of Akron's Department of Computer Science. Do not distribute document copies to the public in any way.

***THIS PROGRAM'S CONTENT IS PROTECTED***

The document and associated content are either the property of, or used with permission by, The University of Akron, Department of Computer Science and may be protected by copyright and other restrictions. Copyrights in the site content may also be owned by individuals and entities other than, and in addition to, The University of Akron, Department of Computer Science. The use of this document by you, or anyone else authorized by you, is prohibited, except for purposes of completing the homework/project. Any unauthorized use of the document may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.