

contains Subject CourseCode (e.g., CSE 310) and the course's instructor email address. The result set must be sorted by CourseCode in ascending order. Print the results for all 200 level courses. [10]

Assignment 8 LINQ Application (Submission Required, 50 points)

Due: April 18, 2020, by 11:59pm (Arizona Time)

2. For the same Excel file Courses.csv file, in this part of the assignment, you will create an XML document from the data source and query the XML data source using LINQ to XML classes. Use comments to indicate what part of code answers which question. You can use a console application or other .Net application template to implement this assignment. You can put the Courses.csv into the App_Data folder in your application. [50 points]
 - 2.1 Define the Course class consisting of at least the following members: CourseId, Subject, CourseCode, Location, and Instructor. Write a program to read the Courses.csv file and create an in-memory **XML** file of the objects of the Course class. [10]
 - 2.2 Save the XML document into the App_Data folder in your application. [10]
 - 2.3 Create the following LINQ queries to extract data from your in-memory XML source.
 - a. Retrieve the list of CPI courses with a course number of 200 or higher. Deliver the result set in the type of IEnumerable<XElement>, where each entity contains course title and course instructor. The result set must be sorted by instructor in ascending order. Print the results. [5]
 - b. Retrieve and deliver the courses in groups in two levels: Use the course subject, (e.g., CPI, CSE, and IEE), as the first level key, and use the course code (e.g., 240, 310, and 494) as the second level key. Print the groups that have at least two courses in the second level group. [10]
 - 2.4 Use the Course XML document and Instructors list (created Assignment 7) as data sources. Write a query to find each course and the course instructor's email address. Deliver the result set in the type IEnumerable<XElement>, where each entity contains Subject and CourseCode (e.g., CSE 310) and course instructor's email address. The result set must be sorted by CourseCode in ascending order. Print the results for all 200 level courses. [15]

General Submission Requirement

All submissions must be zipped into a single file electronically submitted to the Canvas assignment folder. All files must be zipped into a single file for submission.

For programming assignments, the entire solution folder with all the files must be included. To make sure that you have all the files in the zip file, unzip the file on a different machine or in a different directory, and test if you can run the project in a different location.

Submission notice: A programming assignment typically consists of multiple distributed parts. They may be stored in different locations when you create them. You must copy these projects into a single folder for Canvas submission. To make sure that you have all the files included in the zip file and they work together, download your own submission from the Canvas. Unzip the file on a different machine, and test it and see