## CSP2103: Markup Languages

Assignment 1: XHTML, XML, XSLT and CSS Skills Builder

Assignment Marks: 20 / 100

## Assignment Specification:

In this first assignment you will build skills in client side development using xhtml, xml, xslt and css. Specifically, you will build an xml-driven website which allows for the display of student information as seen by university lecturers. You will need to design and build an xml file that will allow you to store the information about individual students (as shown in the example on the following page). You will then design an XSLT sheet which will retrieve this information and display it on screen.

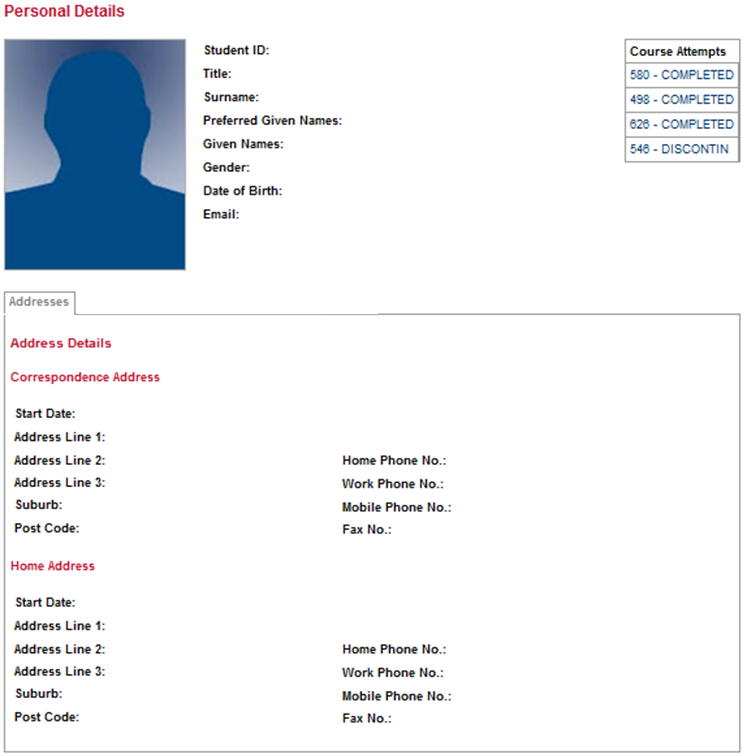
***Task***

Your task is to build an xml document that allows details about courses to be stored, including;

* Student name details
* Student contact details (contact and home)
* Student address details (contact and home)
* Date of birth
* A picture of the student
* Any courses that the student is enrolled in
* Any notes concerning the student

The first part of this assessment will require you to design an XML file that will readily store the information for each given student. Once your XML file is designed and the data entered you will then build an XSLT file which will read the contents of the XML and place it on-screen so it looks as close as possible to that shown on the following page – though you are welcome to enhance the look and feel given everything else is working well.

Your XML file needs to store the details for a minimum of eight (8) students, each with different details, pictures and course attempts. At least two (2) of the students need to have been enrolled in more than one course, either as a Completed or Discontinued or a combination of both. All student records need to load onto the one page one after the other, and the eight records should consist of made up details only (and no joke content).



Essentially the deliverables will four (4) files and a number of additional files.

**XML**

**DOCUMENT**

**XSLT**

**DOCUMENT**

**DYNAMIC**

**XHTML**

**DOCUMENT**

DTD

FILE

CSS

FILE

As well as the presentation of the student information, these are some of the logical requirements of your system;

* Students should appear in alphabetical order according to surname (using a sort function)
* There should be a count function at the bottom of each student record that displays the number of notes and courses listed against each student (which should automatically change as notes/courses are added or removed)
* For course(s) listed against a student’s record, there should be link to the ECU handbook for that course, ie the U67 Bachelor of IT in the handbook here <http://www.ecu.edu.au/handbook/course?id=U67&year=2017>
* A count function at the bottom of the page that displays the total number of student records (which should automatically change as records are added or removed).

Your second assignment will use the first as a base, so your investment of time and effort in this first assignment will pay off in both assignments.

The key to this assignment is completing all of your workshop materials, looking at all of the online examples, reading the indicated chapters of your textbook and asking lots of questions.

***Submission of Deliverables***

Once your assignment is complete, zip the web folder containing your code into a single file called *csp2103\_assign1.zip* or *csp4102\_assign1.zip*. Upload this to the Blackboard assignment submission area on or before the assignment due date and time. Assignments are marked AS IS – if there are strange configuration issues or code errors that cause some or all of the functionality to fail, marks will be subtracted accordingly.

***Academic Integrity***

Web development relies on new developers being able to find examples of code that performs a task (or tasks) that they need, so that they can understand how it works. In many cases, web sites provide freely available code snippets, functions and tutorials. Learning from and using these are perfectly legitimate in your assignment development and your actual assignment, as long as it is done within the university rules. Blackboard contains links to guides on referencing and the university handbook entries on academic misconduct. These will guide you as to how much you can use and how to use it, and it goes without saying that ANY code you use must be fully referenced, and must come only from sites that make it freely available. Any code or interface features found in your assignments that have not been referenced, or have been written by someone other than yourself will be reported as academic misconduct.

Under no circumstances work on your assignment with other students or exchange code with previous or current students. All assignments are crossed-compared during marking and collusion / plagiarism will lead to a report of academic misconduct.

**CSP2103 Assignment 1: Marking Key**

Student No:

Student Name:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Poor | Fair | Good | Excellent |
| **DTD** | (0-1)  DTD did not align to XML document  DTD not correctly linked to XML file  DTD did not read XML structures and elements  DTD did not read XML attributes | (2)  DTD aligned to XML document for the most part  DTD was correctly linked to the XML document  DTD did read some of the XML structures and elements  DTD did read some of the XML attributes | (3)  DTD well aligned to XML document  DTD was correctly linked to the XML document  DTD did read a majority of the XML structures and elements  DTD did read most of the XML attributes | (4)  DTD in perfect alignment with XML structures and attributes |
| **XML** | (0-2)  XML document was not well structured  XML document was not extensible  XML document was not representative of the assignment data requirements  XML content was not relevant to the assignment requirements | (3-4)  XML document showed some evidence of good structure  XML document showed some evidence of extensibility  XML document representative of some of the assignment data requirements  XML content demonstrated some relevance to assignment requirements | (5-6)  XML document was well structured  XML document showed solid evidence of extensibility  XML document representative of a majority of the assignment data requirements  XML content demonstrated solid relevance to assignment requirements | (7-8)  XML showed excellent understanding of assignment requirements, extensibility and the role XML plays in capturing data structure and content |
| **XSLT** | (0-2)  XSLT file contained numerous errors  XSLT used hard-coded or redundant values  XSLT file did not correctly read XML structures or content  XSLT interface was not laid out correctly  XSLT did not display images  XSLT count and sort functions not implemented | (3-4)  XSLT file contained some errors  XSLT used a small number of hard-coded or redundant values  XSLT file correctly read most XML structures and content  XSLT interface featured some good use of layout  XSLT displayed some working images  XSLT count and sort functions partially implemented | (5-6)  XSLT file contained very few errors  XSLT used no hard-coded or redundant values  XSLT file correctly read all XML structures and content  XSLT interface featured very good use of layout  XSLT displayed images correctly  XSLT count and sort functions logically implemented | (7-8)  XSLT showed excellent understanding of the interaction between XML and XSLT in terms of generating a dynamic HTML output with no identifiable errors in a visually pleasing format that fully met or exceeded assignment requirements |

**Final Comments**

|  |
| --- |
|  |
| **Total Mark: \_\_\_ / 20**  **Assignment Grade: \_\_\_** |