

CIS*2750
Assignment 2
Deadline: Tuesday, October 22, 9:00am
Weight: 14.5%

Assignment 2 will consist of three modules. To simplify regression testing of our Assignment 1 code, we will not be updating any Assignment 1 functionality.

Assignment 2 modules:

1. Functions for validating GPXdocs and writing them to a file
2. Functions for computing / retrieving various properties of GPXdocs
3. A set of "glue" functions to convert a GPXdoc and its components into JSON strings. These functions will be useful in later assignments: they will help us integrate the C library created in A1/A2 with the server-side JavaScript code in A3/A4.

You are provided with a temporary header file for A2 Module 1 ([GPXParser_A2temp.h](#)). This header will be updated when Modules 2 and 3 come out. Once Module 3 is released, I will post the final official header file for Assignment 2, which will be used in the A2 test harness for grading.

Module 1 functionality

```
bool validateGPXDoc(GPXdoc* gpxDoc, char* gpxSchemaFile);
```

This function takes a GPXdoc and the name of a valid GPX schema file, and validates the doc against the GPX schema specification. It returns `true` if the GPXdoc contains valid data, and `false` otherwise.

```
bool writeGPXdoc(GPXdoc* doc, char* fileName);
```

This function takes an GPXdoc struct and saves it to a file in GPX format. Its arguments are a GPXdoc and the name of a new file. It must return `true` if the write was successful, and `false` if the write failed for any reason - invalid GPXdoc, invalid output file name, etc.. This function would call `validateGPXdoc` internally to ensure that the GPXdoc argument is valid.

This might seem like a lot of work, but the `libxml2` library has a number of functions that can help you. Most importantly, it has functions for:

- Writing a libxml tree to an XML file
- Validating a libxml tree against a schema file

In both cases, the libxml tree is represented as an `xmlDoc` struct - just like the one you get from `xmlReadFile` in Assignment 1. As a result, both `writeGPXdoc` and `validateGPXDoc` become quite simple once you create a helper function that can convert a GPXdoc into a libxml tree.

As with Assignment 1, the libxml2 documentation has some useful examples that can get you started:

- Creating an XML tree struct - i.e. an XML tree: <http://www.xmlsoft.org/examples/tree2.c>
- Saving an XML tree to a file: <http://www.xmlsoft.org/examples/tree2.c>
- Validating an XML tree against a schema file: <http://knol2share.blogspot.com/2009/05/validate-xml-against-xsd-in-c.html>

Useful documentation (in addition to parser.h on the libxml2 site):

- tree.h contains the functions for navigating and creating XML tree elements: <http://www.xmlsoft.org/html/libxml-tree.html>
- xmlSchemaTypes.h contains functions for working with XML schema: <http://www.xmlsoft.org/html/libxml-xmlschematypes.html>

NOTE: `libxml2` has some issues handling namespaces. To make your life simple, do not try to add the `xmlns` value an actual namespace to an `xmlDoc` or `xmlNode`. Simply add the attribute with the name `xmlns` and value `http://www.topografix.com/GPX/1/1` to the `xmlNode` that corresponds to the `gpx` element, just like you would with the

`version` and `creator` attributes. Do not worry about `xmlns:xsi` and `xsi:schemaLocation` attributes. They do not need to be added to the output GPX file, and their absence will not affect anything.