

# Advanced Programming (CSE 399)

## Homework Assignment 6

Due Friday, February 25, at noon

A collection of files that form the starting point of this assignment is available on the Schedule page under the main course web page. Begin by grabbing these files, unpacking them, and making sure that you can successfully run the main program (`Main.hs`). Either GHC or Hugs can be used for this assignment (GHC is recommended for its better error reporting, but you'll need to invoke it with the `--make` switch so that it knows to look for the HaXml library; the provided `Makefile` does this for you).

Before starting, take a look at `Main.hs` to see what it is doing. Also, read through the definitions of the HaXml combinators in the provided file `Text/XML/HaXml/Combinators.hs` to familiarize yourself with what's there and remind yourself how they work. You may also want to refer to Handout 14 (the slides from class) for intuitions and examples; it is also available from the Schedule page.

1. Use the HaXml combinators to write a function that returns a list of the lines spoken by a given speaker. Make it a valid document by wrapping the results in a root "play" element. (See `Main.hs` for a template of this function and the test file `test-linesofspeaker.xml` for an example of the intended output.)
2. Write a function that deletes all speeches by a given speaker. (Again, see the provided files for template and sample output.)
3. Write a function that adds a speaker node to the top of each scene, listing the names of the speakers appearing in this scene.
4. Write a function that flattens the document by removing all acts and scene nodes, leaving their constituent line nodes. Leave the rest of the document as it is.
5. **Discussion.** (Note: Give some energy to this part — I am really interested in your thoughtful answers. This question will count for 20% of the overall grade for the assignment.)
  - (a) We observed in class that the `CFilter` type be made into an instance of the `Monad` class? Would it be useful to do this?
  - (b) Did you find the fundamental idiom of XML processing using HaXml to be natural and convenient for the purposes of the tasks in this assignment?
  - (c) For what sorts of XML processing tasks would you expect these combinators to be most appropriate? For what sorts of tasks are they most awkward?
  - (d) Can you think of a different way of approaching the problem (i.e., a different basic type of XML transformers to replace `CFilter` and a new set of combinators for that type) that would work better for at least some of these tasks?
6. **Extra credit:** Implement your idea from 5d and test it on some of the transformations from the rest of the assignment.

**Submission:**

- Add your solutions to all parts into the file `Main.hs`.

For question 5, either write your answers in an ascii comment or else email me a separate PDF file (and put a note in a comment in `Main.hs` that you are doing so, so I don't miss it when grading).

If you choose to do the extra credit part, send it as a separate email message, clearly marked.