CS 585 – Domain-Specific Languages
Syllabus for Spring 2018

CS 585, Domain-Specific Languages, is a four-credit course for graduate students in the programming languages sequence. This course presents aspects of the design and implementation of domain-specific languages and focuses on the development of domain-specific languages. A major part of this class will be a student project, in which each student (possibly in a team) will design, redesign, and implement (prototypically) a domain-specific language of their choice. The format of this class will be a combination of (1) presentations by students of their own evolving domain-specific languages and existing, related languages and (2) lectures about language design issues, representations, and implementation techniques.

Prerequisites (from CS 581):
(1) A solid understanding of programming language theory, including syntax and semantics definitions for programming languages, lambda calculus, binding and scope, type systems. It is essential to understand the approach for defining denotational semantics.
(2) A working knowledge of Haskell, including functions, recursion, data types and pattern matching. Students will have to represent abstract syntax and denotational semantics using Haskell.

These prerequisite skills are typically acquired by taking CS 581 prior to this class.

Instructor:
Martin Erwig
Email: erwig@oregonstate.edu
URL: oregonstate.edu/~erwig

Office Hours (KEC 3045):
Monday 11am – noon
Wednesday 3pm – 4pm
or by appointment

Recommended Background Reading:
Haskell: the Craft of Functional Programming, 3rd Ed.
By Simon Thompson, Addison-Wesley, 2011

Learning Objectives:
On completion of the course, students must demonstrate the following abilities.

1. Students will be able to analyze application domains and identify essential objects, operations, and combinators.
2. Students will be able to define syntax and semantics of domain-specific languages.
3. Students can identify opportunities for introducing binding constructs to support abstraction in the design of domain-specific languages.
4. Students will be able to critically review and improve DSL designs.
5. Students will be able to implement prototypes of domain-specific languages in Haskell.
Tentative Lecture Syllabus [(L)ectures & (S)tudent Presentations]:

<table>
<thead>
<tr>
<th>Week</th>
<th>Tuesday Class</th>
<th>Thursday Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td><strong>L:</strong> Introduction, Denotational Semantics in Haskell, Deep &amp; Shallow Embeddings</td>
<td><strong>S:</strong> Identify potential DSL projects (present ≥ 2 DSL ideas with DSL Report Card [Sections 1-4])</td>
</tr>
<tr>
<td>(2)</td>
<td><strong>L:</strong> Domain Analysis, SDDD I: Semantic Design (Domain Decomposition &amp; Modeling)</td>
<td><strong>S:</strong> Select DSL project (present updated DSL Report Card, [Sections 1-4])</td>
</tr>
<tr>
<td>(3)</td>
<td><strong>L:</strong> SDDD II: Syntactic Design (Micro DSL Syntax, Domain-Integration Syntax)</td>
<td><strong>S:</strong> First draft of language design (with DSL Report Card [Sections 5-6]), find example DSL programs</td>
</tr>
<tr>
<td>(4)</td>
<td><strong>S:</strong> Second draft of language design (with DSL Report Card [Sections 5-6]), identify limitations</td>
<td><strong>L:</strong> Cognitive Dimensions Analysis, evaluate student designs</td>
</tr>
<tr>
<td>(5)</td>
<td><strong>L:</strong> SDDD III: Language Operators (syntactic, semantic, first-order, higher-order)</td>
<td><strong>S:</strong> Presentation and critique of related DSL designs</td>
</tr>
<tr>
<td>(6)</td>
<td><strong>L:</strong> Review for midterm</td>
<td><strong>Midterm exam</strong></td>
</tr>
<tr>
<td>(7)</td>
<td><strong>L:</strong> Monadic DSLs &amp; Representation of binding constructs in DSLs</td>
<td><strong>S:</strong> Presentation and critique of related DSL designs (1)</td>
</tr>
<tr>
<td>(8)</td>
<td><strong>L:</strong> Combinator Libraries and DSLs</td>
<td><strong>S:</strong> Presentation and critique of related DSL designs (2)</td>
</tr>
<tr>
<td>(9)</td>
<td><strong>L:</strong> Learning from example DSLs (Haskore, Parsec, ...</td>
<td><strong>S:</strong> Presentation and critique of related DSL designs (3)</td>
</tr>
<tr>
<td>(10)</td>
<td><strong>L:</strong> Representation is key</td>
<td><strong>S:</strong> Final project presentations</td>
</tr>
</tbody>
</table>

Grading

20% Class presentations  
30% Midterm exam  
50% Class project  

Important Note

Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should know, or who need special arrangements in the event of evacuation, should make an appointment with the instructor as early as possible, however, no later than the first week of the term. In order to arrange alternative testing the student should make the request at least one week in advance of the test. Students seeking accommodations should be registered with the Office of Services for Students with Disabilities.

This syllabus and other information can be found on the course home page:
http://eecs.oregonstate.edu/~erwig/cs585/