Syllabus

Bookmark this page

The complete Systematic Program Design course consists of 3 parts, each of which is 4-5 weeks long. In part 2, each week consists of 2 modules, and those modules all have a similar structure, comprised of:

- An overview describing the module learning goals and summarizing the work required to complete the module.
- A number of blended topic lectures, consisting of video interspersed with questions for you to answer.
- A set of problems that will let you practice the new design techniques before the quiz.
- A module quiz. The module quiz is either a set of questions on the week's material, or a self-assessed design problem.
- A module wrap up.

The following chart provides an overview of the course topics:

Week	Module Name	Lectures	Time to complete	Practice Problems	Quiz
	Overall learning goal				
Part 1 of the course covers BSL, the HtDF, HtDD and HtDW Recipes, and Compound Data.					
1	**Self-Reference**	7	5-7 Hours	4	*Multiple Choice Quiz*
	Learn how to use well-formed self-referential data definitions to represent arbitrary sized data.				
	Reference	3	4-6 Hours	2	
	Learn to predict and identify the correspondence between references in a data definition and helper function calls in functions that operate on the data.				
2	**Naturals**	2	3-4 Hours	2	*Self-Assessed Design Quiz*
	Design an alternate data definition for the natural numbers, and learn to write functions using this new data definition.				
	Helpers	6	6-9 Hours	1	
	Learn a set of rules for designing functions with helper functions.				
3	**Binary Search Trees**	6	5-6 Hours	3	*Multiple Choice Quiz*
	Design a data defintion for Binary Search Trees, and learn to write functions operating on BSTs.				
	Mutual Reference	5	6-7 Hours	1	
	Learn to design with mutually referential data.				
4	**Two One-Of Types**	2	3-5 Hours	2	*Self-Assessd Design Quiz*
	Learn to use a cross-product of type templates table to design functions operating on two complex pieces of data.				
	Local	6	8-10 Hours	4	
	Learn to use local expressions in your function designs.				
Part 3 of the course covers abstraction, generative recursion, search, accumulators, and graphs.					

Previous Next