

# CSC 253/453 Collaborative Programming and Software Design

## Fall 2019 Syllabus (August, 2019)

Symphonic	Motifs	Melodic
Principles	Essential difficulty	(Brooks, SF) Software complexity (SF) Think like a computer
	Principles of design	(SF) Modularity by information hiding, structure, transparency, contraction/extension, product families
	Biology/nature/society/knowledge "systems"	Selected readings by Bateson, Dewey, Emerson and others
Pragmatics	Dynamic programming	Anonymous functions, iterators: each, inject
		(Kay) Pure object-orientation, modules
		(Olsen) Design patterns: observer, command, proxy, decorator, singleton, builder, DSL, configuration
		(Matz) Ruby language extension
	Strong static typing	(Steele/Gabriel) Continuations
		(Krishnamurthi) Type fundamentals, parametric polymorphism
		(Lipovaca) Haskell type classes, (Klabnik/Nichols) Rust traits
Meta-programming	(Lipovaca) Haskell functors and monoids	
	(Klabnik/Nichols) Rust error processing	
Collaboration	Software engineering basics	(Matz) Ruby meta-class hierarchy, (Olsen) Singleton/delegation patterns
		(Klabnik/Nichols) Rust meta-programming
		(Schach) Software dev. process, unified process, workflow, phase
		(Schach) Teamwork
	Tools	(Schach, SF) Documentation
(SF) Personal/professional/social responsibilities		
Other topics	Analysis	(Mackall) Distributed version control
	Abstractions	Testing
	Other designs	Memory hierarchy models and analysis (manuscript)
		(SICP) Streams
		Software systems e.g. Rust, LLVM compiler

Brooks: No Silver Bullet – Essence and Accidents of Software Engineering, Fredrick Brooks, 1986.

SF: Software Fundamentals : collected papers by David L. Parnas, Addison-Wesley, c2001. (Chapters 8–10, 14, 16, 26–28)

Olsen: Design patterns in Ruby, Olsen, Russ, Addison-Wesley, c2008. (Chapters 3, 7, 9, 12, 14–18)

Steele/Gabriel: The Evolution of Lisp, Guy Steele and Richard Gabriel, History of PL, 1993. (Section 2.8)

Kay: The Early History of Smalltalk, History of PL, 1993.

Matz: Ruby 1.8.7 code, Yukihiro "Matz" Matsumoto, 2008.

Krishnamurthi: Programming Languages: Application and Interpretation, Krishnamurthi, Shriram, c2003-07. (Chapters 24, 25, 27 – 29) **Department colloquium speaker Monday noon Oct. 14 (fall break)**

Lipovača: Learn You a Haskell for Great Good! A Beginner's Guide, Miran Lipovača, No Starch Press, c2011. (Chapters 6 – 11)

Klabnik/Nichols: The Rust Programming Language, Steve Klabnik and Carol Nichols, 2018 Ed. (Chapters 9, 12, 19.5)

Schach: Object-oriented Software Engineering, Schach, Stephen R., McGraw-Hill, c2008. (Chapters 3, 4, 10–13)

Mackall: Mercurial v0.1 - a minimal scalable distributed SCM (and code), Matt Mackall, 2005.

SICP: Structure and Interpretation of Computer Programs, H. Abelson and J. Sussman and J. Sussman, MIT Press, 1984 (Section 3.5)