

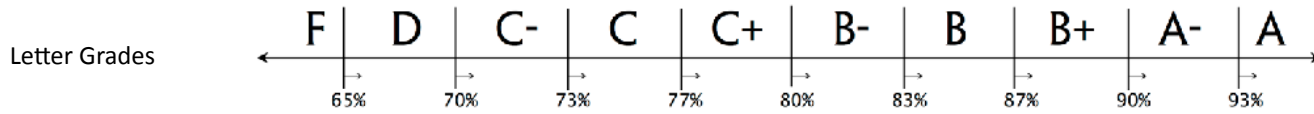
THEORY OF PROGRAMMING LANGUAGES

CMPT 331 • SPRING 2018

-Background

When and where	Wednesdays at 8am and Fridays at 11am in Hancock 2005	
Required Text	<i>Concepts of Programming Languages</i> , by Robert W. Sebesta, some recent edition Published by Addison Wesley – ISBN 978-0-13-139531-2	
Web site	http://www.labouseur.com/courses/tpl	
Instructor	Alan G. Labouseur Hancock 3007 Office hours are posted.	Alan.Labouseur@Marist.edu 845-575-3832 <i>Marist phone</i> 845-440-1102 <i>home office phone</i>

-Grades



You can earn up to 1000 points over the course of the semester, broken down as follows: (These weights are subject to minor variation.)	SE-Radio Episode Summary	10.0%	100 points	[1, 2]
	Programming In The Past	15.0%	150 points	[1, 2]
	Fun with Lambda Calculus	10.0%	100 points	[1, 2, 5]
	Functional Programming	15.0%	150 points	[1, 2]
	Your Own Language	15.0%	150 points	[1, 2, 5]
	Comprehensive Final exam	25.0%	250 points - study sheet permitted	[1]
	Attendance	2.5%	25 points - for consistency	[1]
	Participation	2.5%	25 points - for quality & quantity	[1]
	Laziness adjustment	2.5%	25 points - for not being lazy	[1]
	Whining adjustment	2.5%	25 points - for not whining	[1]

-Objectives and Assessment

Assessment methods include assignments, quizzes, exams, discussions, presentations, peer review, and projects.

[References] refer to Department of Computing Technology Goals available at <http://www.labouseur.com/courses/goals.pdf>

In this course I hope that you will . . .

- learn about and practice programming language criticism based on four domain-in-specific categories and use this knowledge and practice to better understand today's software development environment. [1, 2]
- explore the concepts of many historical programming languages and their impact on the languages of today, remembering that those who forget the mistakes of history are doomed to repeat them. [1, 2]
- avail yourself of the opportunity to develop small programs in many historical programming languages. [1, 2]
- engage in the philosophy of programming languages. [1, 2]
- evolve critical debugging skills by developing programs in many languages. [1, 2]
- enhance your continuing education skills, realizing that capable problem solvers never stop learning. Additionally, preparation and presentation of the projects, as well as participation in class discussions and assignments, requires at least a little research, so there's that to look forward to. [1, 2, 5]
- have fun with programming. [1, 2]

THEORY OF PROGRAMMING LANGUAGES

CMPT 331 • SPRING 2018

-Planned Schedule

#	Week	Chapter	Topic	Due Friday
0	8/29 8/31	1	Introduction • Criteria for programming language evaluation Issues and tradeoffs in programming language design	—
1	9/5 9/7	2	A brief history of programming languages	—
2	9/12 9/14	3	Describing syntax • Fruit flies • Chomsky • Grammars • Sheep Derivations • Parse trees • Grammar ambiguity	<i>SE-Radio Summary</i>
3	9/19 9/21	4	Lexical Analysis and Parsing • Compiler phases	—
4	9/26 9/28	3	Beginning semantics • The need for context-sensitive grammars Attribute grammars • Operational Semantics	—
5	10/3 10/5	3	Operational Semantics • Axiomatic Semantics	<i>Programming In The Past</i>
6	10/10 10/12	3	Axiomatic Semantics <i>No class meeting — Fall break</i>	—
7	10/17 10/19	15	Lambda calculus, part $\lambda f x . (f x)$	—
8	10/24 10/26	15	Lambda Calculus, part $\lambda f x . (f (f x))$	—
9	10/31 11/2	15, 16	Functional programming with LISP and ML Logic programming with Prolog • Unification	Fun with λ Calculus
A	11/7 11/9	6, 5	Data types • Names and binding • Scope • Type checking	—
B	11/14 11/16	9, 10	Subprograms • Parameter passing	—
C	11/21 11/23	—	<i>No class meetings this week — Thanksgiving</i>	—
D	11/28 11/30	13	Concurrency • Threads	<i>Functional Programming</i>
E	12/5 12/7	—	Comprehensive Final Exam in class (likely in two parts) Study sheet permitted; some restrictions apply.	—
F	12/12 @8am	—	Show off your awesome new language.	<i>Your Own Language</i>