

Programming Languages and Paradigms COMP 302

McGill University, Fall 2016

Course Details

Time: Monday, Wednesday, Friday 9:35–10:25

Place: MDHAR G-10

Instructor: Professor Clark Verbrugge

Office: McConnell, room 230

Office hours: Monday, Wednesday, Friday 10:30–11:30, or by appointment.

Phone: 514 398-2411

Email: clump@cs.mcgill.ca

Teaching Assistants: TBA

Email, Website

McGill's MyCourses will be used for course announcements, to manage assignments and for online discussions. Students are expected to monitor their McGill email account for course-related news and information. The external course website is: <http://www.sable.mcgill.ca/~clump/comp302>

Pre-requisites

- COMP 250 (Introduction to Computer Science)

Note: students registering without the pre-requisite may find the course removed from their transcript by their Faculty. Please consult the instructor if you do not have all the pre-requisites.

Textbook

There is no required text for this course. The following optional texts, however, provide useful additional information on many of the topics covered in class:

- *Programming Language Pragmatics*. 4th edition. Michael L. Scott. 2016.
- *Eloquent JavaScript: A Modern Introduction to Programming*. 2nd Edition. Marijn Haverbeke. 2014
<http://eloquentjavascript.net/index.html>
Or any JavaScript reference actually.

Description

This course is designed to provide a thorough introduction to the foundations and paradigms of programming languages. Theoretical and practical aspects of language design and programming practice will be examined in a functional programming context. Topics additionally include basics of programming language design and construction, binding and scoping, parameter passing, lambda abstraction, data abstraction, and type checking.

Evaluation

4 Assignments:	40%
Midterm:	10%
Exam:	50%

A supplemental exam (60%) will be held if required.

In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Assignment and Exam Policy: Assignments must be submitted on time. Late assignments will only be accepted in highly-exceptional circumstances and only with **written** permission of the instructor. No assignment submissions will be accepted after marked assignments have been returned, or after solutions have been discussed in class.

McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offenses under the Code of Student Conduct and Disciplinary Procedures (see <http://www.mcgill.ca/integrity/> for more information).

More specifically, **work submitted for this course must represent your own efforts**. Copying assignments or tests, or allowing others to copy your work, will not be tolerated. Note that introducing syntactic changes into a copied program or assignment is still considered plagiarism.

Assignments

Expected assignment distribution dates and due dates are listed below. Note that this is mainly to help you in general planning; topic descriptions are vague and non-exhaustive, and both the topic and the associated dates may change. Be sure to consult MyCourses for final, official due dates.

Assig.	Main Topic	Available	Due
1	Functional abstraction	Friday, September 16	Monday, October 3
2	Language construction	Friday, October 7	Monday, October 24
3	Evaluation and environments	Friday, October 28	Monday, November 14
4	Procedure abstraction and typing	Friday, November 18	Monday, December 5

Course Content

Lecture dates are approximate and topics will likely shift and/or span lectures. Students are expected to attend all lectures. Missed lecture notes may be obtained from friends, acquaintances, or friendly strangers also taking the class. Readings from Scott's book are indicated below as well. Note that readings are optional, and while they are a good supplement to class instruction they are not sufficient in themselves.

Lec#	DoW	Date	Topic	Textbook Readings (Scott)
1	Friday	September 2	Introduction	1.1–1.5
2	Wednesday	September 7	Course language(s)	
3	Friday	September 9	Functional languages	11.1–11.2
4	Monday	September 12	Functional languages	11.3, 11.5–11.6
5	Wednesday	September 14	Functional languages, Scopes	3.1–3.3
6	Friday	September 16	Scopes, Environments	3.4–3.6
7	Monday	September 19	TBA	
8	Wednesday	September 21	Regular Expressions	2.1.1
9	Friday	September 23	TBA	
10	Monday	September 26	Language design	1.6, 2.1.2–2.1.3
11	Wednesday	September 28	Language design	2.2
12	Friday	September 30	Language design	2.3
13	Monday	October 3	WML definition & execution	
14	Wednesday	October 5	WML definition & execution	
15	Friday	October 7	Control flow	6.1–6.6
16	Wednesday	October 12	Substitution & Invocation	3.7, 9.2.4
17	Friday	October 14	Parameter passing	9.1, 9.3
18	Monday	October 17	Eval, special functions	
19	Wednesday	October 19	Lambda calculus	11.7
20	Friday	October 21	Lambda calculus	
21	Monday	October 24	Impurity, Exceptions	9.4
22	Wednesday	October 26	CPS (optional)	
23	Friday	October 28	Type systems	7.1
24	Monday	October 31	Midterm review	
	Tuesday	November 1	Midterm: 18:05-20:05	
25	Wednesday	November 2	TBA	
26	Friday	November 4	Type systems	7.3, 8.5
27	Monday	November 7	Type systems	10.1–10.2
28	Wednesday	November 9	Objects	10.3–10.4
29	Friday	November 11	Objects	
30	Monday	November 14	Type checking & inference	7.2
31	Wednesday	November 16	Type inference & inference	
32	Friday	November 18	Type inference & inference	
33	Monday	November 21	Semantics	
34	Wednesday	November 23	Dynamic typing	14.1, 14.4
35	Friday	November 25	Virtual machines	16.1–16.2
36	Monday	November 28	Declarative languages	12.1–12.2
37	Wednesday	November 30	Concurrency	13.1–13.2, 13.4
38	Friday	December 2	Esoteric languages	
39	Monday	December 5	Exam review	