Organization of Programming Languages

“OPL”

CS 3363 – Online

CRNs &

Spring 2023

Instructor Information:

Richard L. Churchill,
Office: TBD and by phone. Also, you can arrange one-on-one Zoom meetings with the instructor, or in-person if required.

Class meetings: **ONLINE, days and times TBD.** Students will vote on an agreed schedule via discussion board.

Phone: 405-334-7674
NOTE: I do not respond to text messages. This is due to FERPA restrictions, making my responding to such messages a violation of Federal law. Use email if you need me to read something. Also, only call me if it is important, as email is the preferred form of communication: it allows me to budget my time across all students better.
E-Mail Address: richard.l.churchill@okstate.edu. ***Use of the Canvas email system has been more reliably immediate in the past, so please use that link in Canvas.***

Office Hours: TBD. One-on-one and in-person meetings with the instructor can be arranged. Contact the instructor to schedule such meetings.

T.A. Information:

TBD, MSCS ?.

office hours: TBD

Emails from and to the instructor:

The instructor will use email and Canvas to notify the class and individual students regarding any information about the course at any point in the semester, as well as using in-class announcements. Students are responsible for the contents of these emails and announcements. The email addresses used will be those used on Canvas, which are OSU email addresses. ***Emails to the instructor should be via the Canvas email system, as this reports emails both in Canvas and via the OSU email system.*** Be sure you can receive email from your **okstate.edu** account addresses.

The instructor will endeavor to use subject lines starting with “CS 3363 OPL:” or similar text for all emails regarding this course. Please use that same subject/title for emails you send to the instructor and other students in this course, so that recipients can spot important emails. Allow at least a day for a response to any email to the instructor. Complex questions may require somewhat longer due to any research that may be required.

Textbooks:

- Mandatory text
  - ***Programming Language Pragmatics***, 4th edition, Michael L. Scott, ISBN: 9780124104099.
- Suggested supplemental text
  - ***Elements of ML Programming***, ML97 edition, Jeffrey D. Ullman, ISBN-10: 0-13-790387-1.
  - *This is an older book that is much easier to find as a used book. It is not necessary, but if you have never used a functional programming language it is useful in that it does a reasonably good job of introducing this paradigm. It has become somewhat scarce with the current lockdowns, but copies are available for less than $11.00 as this syllabus is being written. Other languages used in this course do not have recommended texts and you are free to choose any you find that fits your need. Links to on-line tutorials and texts for those other languages will likely be sufficient for the course's needs.*

Non-programming assignments will be largely from the mandatory text. The majority of the instructor's slides will also reference the mandatory text. The text in print format may be ordered from any of a number of on-line retailers. Consider using **addall.com** to search for a better price, as this Web site searches over 30 online sources for availability and price information. Have the text by the middle of the third week of the semester. The first assignment requiring it is due at the end of the third week.

There will be two assignments in ML and additional assignments in LISP, Haskell, Ruby, Matlab and either Prolog or an alternative such as Erlang. There is no recommended text for languages other than ML. Tutorial material on each of these languages will be posted to Canvas, along with links to material online.

LISP, Haskell, Ruby, Matlab, Prolog and Erlang are all languages worth knowing. You may wish to find good, inexpensive books on these languages at some point in your career. Haskell is a language with some very interesting features and is used extensively in several areas of computer science research. Ruby has a distinctive approach (a goal of the designer was "no surprises") and should broaden your understanding of the decisions that may be made in the design of languages. Matlab is a powerful toolset for mathematical problems. Erlang sees extensive use in telephone and networking systems, as well as being implicitly concurrent.

Note that access to the csx server will require use of Virtual Private Network (VPN) software available via the computer science department Web site at http://cs.okstate.edu/loggingon.html. If you have problems accessing the server using the VPN software, contact the department's technical support via http://cs.okstate.edu/techsupport.html. Note also the reference to PuTTy as a useful tool for accessing the server, and that an FTP utility (such as Core FTP Lite) can also be useful.

Prerequisites for This Course:

Prerequisites for this course include experience programming in at least one commonly used programming language, such as Python, C, C++ or Java. Preferably, you should have completed Programming I and II courses at a college or university. Ideally, you should have taken Theoretical Foundations, but this is not required. If you have questions about your experience, ask the instructor.

Course Objectives:

This course is primarily about the decision process in selecting programming languages for tasks and the decisions that must be made in designing a programming language. It is a course that is more about the "why" than the "how."

In this course, we will

- Describe fundamental concepts of programming languages.
- Examine the design issues of various language constructs
    - Examine implementations of these constructs in some of the most common programing languages, and
    - Compare design alternatives.

Roughly speaking

- We study concepts underlying high-level programming languages and different programming paradigms.
- We study how and why different programming languages were developed and their similarities and differences.

Skills and knowledge obtained in this course:

1. Increased capacity to express programming concepts;
2. Background for choosing appropriate languages and language abstractions for solving a given problem;
3. Increased ability to learn new languages, to understand the significance of programming language implementations.
4. An understanding of the applicable domains for languages, including such issues as program efficiency, computational security and reliability.
5. An understanding of the issues and tools for concurrent and parallel programming.
6. The impact of concepts such as type models and exception handling, including on system performance and security.

Style/Mode of Teaching:

This is an online course. Meetings will be held each week. You are required to attend each meeting ***OR*** view the recordings of the meetings you do not attend. This is important since you will be held responsible for all material presented during the online class meetings. Videos of the meetings will be initially placed on Canvas, but due to space limitations all but the most recent recording will be placed in the course's csx account. You will also be provided with links to those videos for you to download.

The assignments are accessed by using Canvas. Non-programming assignments are to be submitted via Canvas drop-boxes, while programming assignments are to be submitted via 'handin' on the department's csx server. Instructions about submissions will be included with program assignments.

The Canvas home page is accessed via http://oc.okstate.edu, logging in to Canvas, then clicking on the course link, or via canvas.okstate.edu. The class assignments in .pdf format will be placed in the Files section as well as being listed in the Assignments area. All dates and times when assignments are due will be posted on Canvas. A preliminary schedule of assignments and exams is located at the end of this syllabus, though it will not appear if viewed directly on Canvas. Feel free to download the syllabus file from the Files / Syllabus folder and use your copy to keep notes on. This will let you see the course calendar. The calendar may be updated online as needed, but will be announced via Canvas.

Graded Materials:

Non-programming Assignments (8 @ 25 points)	200
Mid-term Exams (2 @ 50 points each)	100
Final Exam (1 @ 50 points)	50
Programs (6 @ 25 points)	150
(Extra Credit / Proof of Access is 10 pts)	
Total (for required assignments and exams)	**500**

Assignment Due Dates:

All assignments will be due at 11:59 PM, Stillwater time, on the Saturdays they are due. The first homework assignment does not require the textbook. Programming and non-programming assignments alternate, week to week. There will be no assignments due during Spring Break.

Assignments cannot be made up if they are missed except under circumstances such as severe illness, family emergency, etc. Contact the instructor if the need to submit an assignment late arises. Such contact should be prior to the deadline for the assignment in all but emergency situations.

There will be six (6) programming assignments worth 25 points each for this course, plus one “proof of access to csx” extra-credit assignment for ten (10) points. Apart from the “proof of access” assignment, these will usually be posted to Canvas about two weeks prior to the due date. Programming assignments must be submitted via ‘handin’ to the department’s csx server unless otherwise stated in the assignment document. Late programs will be accepted at a 20% penalty for each day late, with any fraction of a day counting as a full day. Thus, a program four days and one second late would count as five days late, and receive no credit. Non-programming assignments must be submitted to the corresponding Canvas folder.

Acceptable Formats for Assignment Submission:

All non-programming assignments must be submitted in digital formats: an Adobe Acrobat .pdf file, or Microsoft Office Word document format (.docx or .doc) are preferred. Do not submit scans of hand-written text. You may submit scanned images of graphs, diagrams and such embedded in your submissions, but text must be submitted in one the file format listed above. You are in a computer science course. Use the tools that you will be using when you have a job in this field. Microsoft Office is available via the IT department’s Virtual Lab.

Programming assignments must be submitted as properly formatted and documented files of the type required by the language used in the assignment. The submitted files must be named as specified in the assignment. The programs must compile (or interpret) and execute correctly on the Computer Science Department's csx server. NO POINTS WILL BE AWARDED FOR PROGRAMS THAT DO NOT COMPILE/INTERPRET, OR DO NOT RUN, ON THAT SERVER.

Program documentation must include your name, the assignment name, the course and semester it is for, the instructor's name, the date of submission, the purpose of the program, and all other suitable comments regarding the code contained. A documentation template for each programming language will be provided on the Canvas site.

Grading Policy:

Grades will be assigned based on point totals as follows:

A:	90-100%
B:	80-89%
C:	70-79%
D:	60-69%
F:	< 60%

Examinations:

There are **TWO Mid-term Exams** for this course. The mid-terms will be administered Monday, February 28, and Monday, April 11. The **Final Exam** will be on the date specified by the Registrar's office.

Students with disabilities may obtain additional time to complete exams by first obtaining the required endorsements from Student Disability Services. See the Web page at following URL for contact information:

https://apps.okstate.edu/directory/osu/index.php/module/Default/action/ViewFunctionalGroup?group_id=870

Hardware/Software Requirements:

Students are required to have access to a computer and the Internet, to be able to read standard Microsoft Office documents (.docx, .xlsx and pptx files) and .pdf files, and to be able to log into and use the Computer Science department's server (csx#.cs.okstate.edu where # is the cluster to be used). They must also be able to submit documents of the appropriate types for all assignments.

Class Attendance:

Attendance of or viewing of recordings of meetings is required unless otherwise excused. You are responsible for any material presented during meetings.

Quizzes:

There are no quizzes for this course unless the instructor determines attendance has become a problem. The instructor may then add any number of quizzes he so chooses, and they will count towards the point totals required for the various grades.

Projects/Programming Assignments:

Programming assignments are due at the date and time posted in the assignment documents.

Programs are severely penalized for being late. The penalty for late program submissions is 20% per day. See above in Assignment Due Dates for more details. Note, though, that getting an assignment in at all is better than not submitting it. You gain nothing from doing nothing.

Collaboration Policy:

Assignments and Programs: Discussion with classmates and others is allowed, and you should feel free to use the Canvas discussion board. You may not share code. After any such discussion, each student must write up his/her own solution, implementing his/her own program for programming assignments. Copying the work of anyone else, whether a fellow student or any other party, is not allowed and is regarded as cheating. Giving another student your work is considered cheating as well.

Students who do not comply with the described collaboration policy will be reported for an Academic Integrity violation to the University and a punishment up to and including F! will be assessed for this course. University punishments include possible suspension and expulsion.

Disabilities Act:

If you feel that you have a disability and need special accommodations of any nature whatsoever, the instructor will work with you and the Office of Disabled Student Services, 315 Student Union, to provide reasonable accommodations to ensure that you have a fair opportunity to perform in this class. Please advise the instructor of such disability and the desired accommodations at some point before, during, or immediately after the first scheduled class day.

Assignment and Test Dates for Spring 2023 semester:

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	WEEK
JAN	15	16	17	18	19	20	21	1
		Holiday					*HW1*	
	22	23	24	25	26	27	28	2
							P0	
	29	30	31	1	2	3	4	3
							HW2	
FEB	5	6	7	8	9	10	11	4
							P1	
	12	13	14	15	16	17	18	5
							HW3	
	19	20	21	22	23	24	25	6
							P2	
	26	27	28	1	2	3	4	7
			MT!				*HW4*	
MAR	5	6	7	8	9	10	11	8
							P3	
	12	13	14	15	16	17	18	9
		Spring Break						
	19	20	21	22	23	24	25	10
							HW5	
	26	27	28	29	30	31	1	11
							P4	
APRIL	2	3	4	5	6	7	8	12
							HW6	
	9	10	11	12	13	14	15	13
			MT2				*P5*	
	16	17	18	19	20	21	22	14
							HW7	
	23	24	25	26	27	28	29	15
							P6	
	30	1	2	3	4	5	6	16
							HW8	
MAY	7	8	9	10	11	12	13	17
	14	15	16	17	18	19	20	18
				Final Grades				