

Syllabus for CSC 301:

Fundamentals of Programming Languages

University of Rhode Island – Online Course – Summer 2019

Instructor David H. Brown
Office Location Tyler 129
Telephone [401-874-4223](tel:401-874-4223) (email is better; I'm rarely in my office in the summer)
Email david_h_brown@uri.edu (also for Skype/Google)
Office Hours **by appointment**, online or at Tyler Hall
Skype [david_h_brown_uri](https://www.skype.com/people/david_h_brown_uri) (*monitored only during office hours for online courses*)
Class Days/Times course to be completed independently; no fixed online meeting time

TA: Madhukara Kekulandara, mkekulandara@uri.edu

Because of my temporary status, I cannot assign grades of "Incomplete" except in cases of real emergency, in which cases it must be approved by the department chairperson.

Course Description

LEC: (4 crs.) Organization of programming languages, data and control structures, syntax and semantics, compilers and interpreters. Block structured languages, recursion, parameter passing, run-time storage management. Procedural, functional, object-oriented, and logical languages. (Lec. 3, Lab. 2/Online) Pre: CSC 212.

Courtesy of Professor Lutz Hamel whose course design I largely follow:

Language enables thought. In this course we study a class of formal languages known as programming languages. Similar to natural languages, these formal languages enable us to reason about algorithms and procedures to solve computational problems on computers. However, their formal nature restricts the kind of meanings particular language constructs can assume and therefore makes them amenable for the execution on a computer.

Over the years many different programming language dialects have evolved to address particular technical issues, e.g. object-oriented languages, real-time languages, database query languages, logic languages, etc. Here we study the major structures of modern programming languages. Understanding not only the syntax of a language but also the semantics and implementation techniques of this language will allow you to design better programs. Having deeper insights into the design of a programming language will also enable you to learn new programming languages much faster. Having a thorough understanding of today's languages allows you to design the programming languages of tomorrow.

Objectives

Upon completion of this course...

- You will be able to discern and contrast major programming language paradigms in use today.
- You will be able to pick an appropriate language for the job at hand.
- You will more easily adapt to new-to-you programming languages
- You will have deeper insight into the evolution of programming languages.

IDEA course objectives

These IDEA objectives relate to this course, listed roughly in descending order of importance (IDEA is the online course survey we do at the end of the semester):

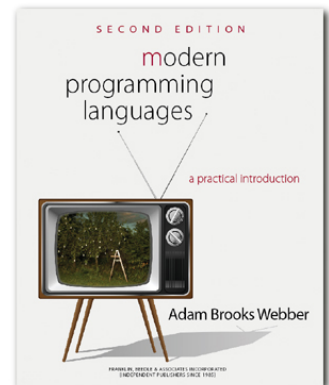
- **Learning fundamental principles, generalizations, or theories:** you will progress toward this objective by attentively attending the lectures and reading the textbook, asking questions, and reviewing the posted lecture slides
- **Learning to apply course material:** the assignments are your primary opportunity to make this happen; please do them. If you find them challenging, talk to me and the TAs!
- **Gaining factual knowledge:** you will find new terminology and new understanding of jargon you've already encountered
- **Acquiring an interest in learning more** by asking questions and seeking answers: I particularly welcome your questions about and comparisons to programming languages you actually use in your work, studies or hobbies. Sharing these in class can really help everyone relate these theories to applied programming

Textbook (required)

Modern Programming Languages: A Practical Introduction, Adam Brooks Webber, Franklin, Beedle & Assoc., 2nd Edition. ISBN-13: 9781590282502

Author's site: <http://www.webber-labs.com/mpl.html>

Online edition: <https://www.redshelf.com/book/16544/modern-programming-languages-2nd-edition-16544-9781590282694-adam-brooks-webber>



Why it's worth your money

You can develop a deeper understanding from a second perspective. The lecture slides and my videos do not cover all the material and will necessarily emphasize different aspects. You should first skim through the chapter to be familiar with the topics, then work through the slides and any videos, then read the chapter carefully. Once done, you should be very ready to tackle the assignments and apply your understanding to the more challenging exam questions.

Develop mastery of the new programming languages and techniques. It takes practice, and the assignments we grade are not enough for you to be able to comfortably solve coding problems in the timed exams. Several of the chapters include numerous example exercises which you can use to hone your skill.

I will place a copy on 2-hour reserve in the Kingston campus library. The first edition is very similar and could be used in a pinch, but there are some differences. Helpfully, the 2nd edition shows output from more current versions of the programming language environments we use. There are a few errata listed on the author's site.

Check Sakai for readings to prepare for class ahead of time. Lecture slides will also be posted (different from the author's which are available at the above URL, but similar).

Accessible textbook

The publisher offers the textbook for online access through RedShelf who claim that their reader works with a variety of assistive technologies: <https://about.redshelf.com/accessibility>

Required Technology

You will need a reliable **broadband Internet connection** and a personal computer with a modern, **standards-compliant browser** to access the course. See information at <https://sakai.uri.edu/> and http://hdwiki.uri.edu/index.php/Student_Sakai

You may need additional software (plug-ins or separate programs) to access different types of content, such as **PDFs** and **videos**. You may need to reduce security settings for the uri.edu domain such as allowing **pop-up windows**.

Browser hint: Many browsers will go to the previous page if you hit “Backspace.” This can be devastating in Sakai if you’ve typed a long response to an exam question and lose it all. In Firefox, go to the address `about:config` and set `browser.backspace_option` to `1` instead of `0` to turn off this “feature.” Chrome is expected to turn off this behavior in version 52 at the end of July ‘16, but you can get extensions to change/fix the behavior sooner. Don’t know about Safari/IE/Edge/Opera.

You must be able to install software on your **personal computer** (a notebook is fine; you will need a keyboard). Throughout this course we will be using various programming language and software development environments including: SML, Java, and Prolog. All free to use and available for Mac OS, Windows, and Linux. More details will be given on Sakai. You will generally need administrator privileges on your computer to be able to install the software. The computer you use does not need to be particularly powerful to use this software.

You will need **a way to scan or photograph** hand-drawn diagrams for a few of the assignments and for some exam questions:

- A scanner such as is often built into all-in-one printers
- Your smartphone’s camera combined with an app like **Microsoft’s Office Lens** (free for iOS, Android, and Windows Phone) – the app “de-skews,” removes the background, and enhances contrast (works well for whiteboards in your face-to-face classes, too)
- Any camera if you can make sure the work is well lit and photograph it straight on, not at an angle. Pinning it to a bulletin board can work well.
- You may also choose to use a computer drawing program, but that generally takes more time.

Expected skills

This course’s prerequisite, CSC 212, is titled Data Structures and Abstractions. CSC 212’s prerequisite is CSC 211, Object-Oriented Programming. From these prerequisites, you should be comfortable with the concepts of pointers, linked lists, and stacks; functions, iteration, recursion, classes and program structures. If you can write both iterative and recursive implementations of an in-place array-based Quicksort, that is great. If not, you should at least be able to read and understand code of that complexity.

You will need to have or develop a minimal degree of **comfort with the Command Prompt** (Windows), Terminal (Mac OS) or Shell (Linux) on your system to enter commands and navigate the directory tree (`cd`, `pwd`, etc). This will enable you to effectively use the free software (programming languages and related tools) required for this course on your own computer. You will want administrator privileges to be able to install software yourself.

You must have sufficient **command of your computer’s interface** (including keyboard/mouse skills or the equivalent) to complete work within a reasonable amount of time. Most handwriting and voice recognition systems are not well suited to programming.

You must be able to **maintain your focus** on and engagement with the coursework without the external prompt of a class meeting. This course includes the full content (plus required discussion forums) of the regular Fall/Spring version of the course compressed into 10 weeks instead of 13. Taking the course online makes it harder, not easier; taking the course in the summer makes it harder, not easier.

You must **be willing to ask (and help answer) questions** in the forums. Everyone in the class is still learning about these topics; do not be shy! I have yet to hear or read a “bad” question. Occasionally, questions can be off-topic or tangential, but that’s okay (especially online where we aren’t as limited in time and space). Remember, you’re graded on your participation

in the forums. However, you must not post solutions to assignments – or attempted solutions: not even code that doesn't work. Try posting just an error message; someone has probably seen it before. You *can* email your code to me or the TA for help – another good reason to begin assignments early! (I prefer that you copy and paste the code, including any error messages into the body of your email, attached source files just add steps to read them... steps not available if I'm out of the office using my phone.)

Grading

Your work will be evaluated in five categories

Category	Weight
Homework and Programming Assignments	30% (varying point values)
Exams	50%
Labs	5%*
Forum discussions	5%*
Quizzes	5%*

The total weight specified is 95%. The remaining 5% will be taken from whichever of labs, forum discussion, or quizzes you individually do best at. So, if you get 80% on the labs, 70% on the forums, and 90% on the quizzes, labs and forums will each count for 5% of your grade and quizzes will count for 10% of your grade.

Percentages will be rounded to the nearest whole number and letter grades assigned in these ranges: **A** 94-100, **A-** 90-93, **B+** 87-89, **B** 83-86, **B-** 80-82, **C+** 77-79, **C** 73-77, **C-** 70-72, **D+** 67-69, **D** 60-66, **F** <60.

Because of my temporary status, I cannot assign grades of "Incomplete" except in cases of real emergency, in which cases it must be approved by the Department Chairperson and Dean.

Homework and Programming Assignments

Most of these are based on small subsets of the end-of-chapter exercises – just two or three, typically. However, there are often specific instructions to follow that are given in the related PDF description. Make sure to follow any instructions carefully. You must develop the solutions to the exercises yourself using methods and techniques from the course material. Do not use solutions you may find online; that's academic dishonesty... and they probably are using language features we haven't which makes it easy to spot. See the general rule for assignments near the top of the syllabus.

You will benefit immensely by doing additional exercises from the book. In particular, I recommend you work through chapters 7 and 9 which have some good problems for learning to write ML programs; chapter 19 does much the same for Prolog. Exam questions will include writing short programs in these languages, using recursion wherever appropriate. The more you've done, the easier this will be for you.

The homework assignments are weighted individually in a range from 5 to 15 points for a total of 110 points.

Exams

Each Exam will be available in Sakai for a period of at least 24 hours during which you must complete it. Sakai will "retract" the exam automatically at the end of the exam period, so begin well before the end of the exam period or you will not have the full time available to you.

You will be able to save and go back to earlier questions but you will not be able to pause and resume later; you can submit the exam only once. I strongly encourage you to write your responses in a plain text editor and copy/paste into Sakai. If something horrible happens with the web site like you accidentally close your browser, you could at least email me that file (make sure to do so before the exam period expires).

Where the response to an exam question is most easily answered with a hand-written diagram, you will be able to upload a file, e.g., a picture taken with Office Lens or scanned image. Whatever method you use, practice and test it before you begin the exam to be sure you don't lose time fiddling with software during the exam.

Discussion participation

I hope you will make full use of the interactive possibilities of the online course such as group chats and the support forums. There are four required discussion forums (weeks 1, 3, 6, and 8). Typically, you will be required to post your initial thoughts on some topic and then later respond constructively and meaningfully to two other students.

Quizzes

Most chapters will have a short, auto-graded question in Sakai to answer. A very small part of your grade, these are primarily intended to help keep you on schedule in the readings. Nevertheless, they carry enough weight to affect your final course grade, so do them!

Labs

The labs are intended to be completed collaboratively with other students during the week. That's going to be difficult with an online course; we'll see what we can do with online interactions. After you complete the lab, there is a short assessment in Sakai to complete for your lab credit. We might skip a week, but expect a lab weekly; see Sakai.

Course Policies

- **Check the Sakai website often!** You should receive automatic notifications from Sakai; if you don't, you must fix that and make it happen. Usually, this means that you must have an email address you will read at least daily set up in Sakai. I will try to keep the website as up-to-date as possible.
- Class **attendance, promptness, participation, and adequate preparation** for each class are expected. If you are absent, it is your responsibility to find out what you missed (e.g. handouts, announcements, assignments, new material, etc.)
- **Late assignments** may, when the course schedule allows, be accepted up to 1 day late with a penalty of 25% of the points available. Do not leave the assignments for the last minute or even hour. Most are trickier than they seem and Sakai's clock might be running fast and close the assignment before you expect. Special arrangements *may be made* with the instructor in the event of a prolonged illness or other valid / documentable reason.
- **Make-up quizzes and exams** will **not** be given without a valid excuse, such as illness. If you are unable to take a scheduled examination due to valid reasons, please inform me prior to the exam time. Under such circumstances, you are not to discuss the exam with any other class member until after a make-up exam has been completed.
- **Do not share program code or assignment answers.** All work is to be the result of your own individual efforts unless explicitly stated otherwise. See also the Academic Honesty statement below. I do encourage you to share your questions, your insights, and your own understanding with other students... but never your files or responses to assignments (or exam questions).
- **Instances of apparent plagiarism, unauthorized cooperation or any form of cheating** will be reported to your Dean (including non-URI students), my Dean, and the Dean of Students as required by section 8.27 of the University Manual. In my brief report, I will identify the student(s) involved and the information I have that leads me to suspect academic dishonesty. (It is not my job to investigate, uncover chains of events, or determine "guilt." I cannot give you a copy of my report as it may include other students' confidential academic records.) Generally, I will record a 0 for the assignment and not request further action be taken. I will email you a day before I submit my

report and would be happy to include any explanatory statement you might wish to provide. See also the Academic Honesty statement below.

One other thing... if you fail one of the first four assignments, don't give up: talk to me. You're not the first student who has not caught on to these abstract ways of thinking about programming languages right away. I am willing to average your scores between the original failed assignment and a "redo" assignment under the following conditions:

- Equivalent assignments are available only for the first four assignments
- You may do this only once
- You must make your request within a week of receiving a failing grade (<60%)
- You must email me the equivalent "redo" assignment within a week of receiving it from me

If you exercise this opportunity, do make sure to meet with me or a TA to ensure you really understand the material and can "ace" the replacement assignment.

Summary of Course Schedule

See the course site for more details and any updates: CHANGES ARE POSSIBLE. The pace and scheduling of the course will require you to remain engaged and active throughout the week; it is not possible to do the work only on weekends.

While the course is online, it is not self-paced: assignments are due at specific times; exams must be taken within given windows of time. So, if you decide to go on vacation during these ten weeks and are not able to bring a computer with you and maintain connectivity to complete the work on time, your grade will suffer. Accommodation for religious observation, illness, or participation in University-sanctioned events will be made according to University policy:

<http://web.uri.edu/manual/chapter-8/chapter-8-4/>

Week	Begins	Chapters	Assignments, Quizzes, and Exams (M W F)			Discussion Forum Topics, Calendar notes
1	May 20	1-3		#1 (ch 2); Q2	#2 (ch 3); Q3	Introductions , Paradigms, Syntax (Grammar) Thursday: drop w/ 100% refund
2	May 27	3-4	#3 (ch 4); Q4		Exam 1	Syntax (Precedence, Associativity), Language Systems Monday holiday; work due Tuesday
3	June 3	5-7	Q5	#4 (ch 5); Q6	#5 (ch 6); Q7	ML (Functional Programming) , Types Friday: last day to drop
4	June 10	7-10	#6 (ch 7); Q8	Q9	#7 (ch 9)	ML (Pattern matching, Partial Evaluation), Polymorphism
5	June 17	10, 12; begin 13, 15, 17	Q10		Exam 2	Scope, Memory, Java/OOP (Java / OOP is not on Exam 2)
6	June 24	14, 16; continue 13, 15, 17	Q12 (Tue)	Q14	#8 (ch 17); Q16	Memory, Object-Oriented Programming (Java / OOP is needed for assignment 8)
7	July 1	18 (19)	#9 (ch 18); Q18		Exam 3	Parameters (also begin looking at Prolog) Thursday holiday; no effect on schedule
8	July 8	19, 20		Q19, #10 (ch 19)	Q20	Prolog (Fundamentals), Language grab-bag
9	July 15	20, 22, 23		#11 (ch 20)	#12 (ch 22)	Prolog (Unification, computation), Semantics
10	July 22	23, 24	Q23	#13 (ch 23)	Final	Semantics , History

Assignments and quizzes: "#x (ch y)" means assignment x is based on chapter y (more-or-less) and is due this day. "Qz" means the short, auto-graded quiz for chapter z is due this day. (There are no quizzes for chapters 1, 11, 13, 15, 17, 21, or

24.) **Topics** with a **highlight** are required/graded discussion forums (see online for more information). **Boldface** topics have a support forum.

Each **exam** covers one part of the course, but the final exam is also cumulative. Exams are online in Sakai. The first three exams will be open from 5am Friday until 11am Saturday (a 30-hour window). All work must be complete by the end of the last day of the session, so the final exam will be open from 5:55pm Thursday until 11:55pm Friday (a different 30-hour window). Plan to begin with at least as much time as is allowed before the window closes and your exam is automatically submitted, probably without whatever response you were working on.

Which exam	How long	When	What
Exam 1	1 hour	End of week 2	Part I: Chapters 1-4 (Syntax, Language Systems)
Exam 2	1 hour	End of week 5 ¹	Part II: Chapters 5-9 (ML, Type, Polymorphism)
Exam 3	1 hour	End of week 7	Part III: Chapters 10, 12-18 (Scope, Memory, OOP)
Final Exam	2.5 hours total (three sections)	Last day of session	Part IV: Chapters 19,20,22 (Prolog) Part IV: Chapter 23 (Semantics) Parts I-III: Cumulative

The exams may include some content from the book that is not in the slides, but the slides are a good guide. Do not neglect the textbook if you wish to excel in this course! (See comments about the textbook above.)

Where we deviate significantly from the book

The Java/OOP chapters (13, 15, 17)

Review chapters 13, 15, and 17 early in the course to see what you already know: assignment #8 requires you to modify / extend some Java code. If you are already very familiar with Java and its object model, you won't need much time. If you are coming from a different object-oriented language (e.g., C++, others), you will want more time and may want to begin earlier than week 5. There is a **Java support forum** available.

Chapter 14 and Memory Management

Focus on the slides; we approach these topics more from the perspective of an application programmer. The book develops details and algorithms involved in managing memory, especially the heap. As these routines are a hidden part of any programming language, knowing their secrets are of little practical use unless you are designing your own language.

Chapters not covered

There are two chapters we do not cover at all in the "lecture" slides and assignments: Chapter 11 (A Fourth Look at ML) and Chapter 21 (Cost Models). I do wish we had time for Chapter 11; ML has a great type system. The textbook almost always offers more depth than the slides, especially the chapters on Java and Object-Oriented Programming (OOP). (URI students will already have taken CSC 211, our "Object-Oriented Programming" course, or its equivalent.)

General University of Rhode Island Policy

The following statements of URI policy are recommended for inclusion in all syllabi.

Illness Due to Flu

The nation is experiencing widespread influenza-like illness. If any of us develop flu-like symptoms, we are being advised to stay home until the fever has subsided for 24 hours. So, if you exhibit such symptoms, please do not come to class. Notify me at david_h_brown@uri.edu of your status, and we will communicate through the medium we have established for the class. We will work together to ensure that course instruction and work is completed for the semester.

¹ Based on when we have completed the content for Exam 2, we should probably have the exam on Wednesday or even Monday. But one summer, a student or two didn't notice and thought it was on Friday like the others, missing it. So, we'll put it off until Friday for consistency... just try not to let the new content we'll have covered push the old out of your mind!

The Centers for Disease Control and Prevention have posted simple methods to avoid transmission of illness. These include: covering your mouth and nose with tissue when coughing or sneezing; frequent washing or sanitizing your hands; avoiding touching your eyes, nose, and mouth; and staying home when you are sick. For more information, please view www.cdc.gov/flu or www.cdc.gov/flu. URI Health Services web page, www.health.uri.edu, will carry advice and local updates.

Academic Honesty

Students are expected to be honest in all academic work. A student's name on any written work, quiz or exam shall be regarded as assurance that the work is the result of the student's own independent thought and study. Work should be stated in the student's own words, properly attributed to its source. Students have an obligation to know how to quote, paraphrase, summarize, cite and reference the work of others with integrity. The following are examples of academic dishonesty.

- Using material, directly or paraphrasing, from published sources (print or electronic) without appropriate citation
- Claiming disproportionate credit for work not done independently
- Unauthorized possession [of] or access to exams
- Unauthorized communication during exams
- Unauthorized use of another's work or preparing work for another student
- Taking an exam for another student (or allowing anyone else to take an exam for you)
- Altering or attempting to alter grades
- The use of notes or electronic devices to gain an unauthorized advantage during exams
- Fabricating or falsifying facts, data or references
- Facilitating or aiding another's academic dishonesty
- Submitting the same paper for more than one course without prior approval from the instructors.

Disability

Any student with a documented disability is welcome to contact me as early in the semester as possible so that we may arrange reasonable accommodations. As part of this process, please be in touch with Disability Services for Students Office at 330 Memorial Union, 401-874-2098 (<http://www.uri.edu/disability/dss/>) or 239 Shepard Building, Feinstein Providence Campus, 401-277-5221.

Academic Enhancement Center

This is a challenging course. Success requires that you keep pace with the work, understand course concepts, and study effectively. The Academic Enhancement Center (<http://www.uri.edu/aec/>) is a great place to do this. At the AEC you can work alone or in groups, and tutors and professional learning specialists are available to help you to learn, manage your time and work, and study well. On the Kingston campus, it's open Monday through Thursday from 10 a.m. to 9 p.m. and Fridays until 1 p.m. All services are free (the coffee is free as well!), and no appointment is needed. You can call for complete information at 874-2367, or just stop by the center on the fourth floor of Roosevelt Hall. In Providence, the Academic Skills Center (ASC) is at 239 Shepard Building, (401) 277-5221. Hours are posted each semester at <http://www.uri.edu/prov/studentresources/help/academicskills.html>. In addition, the Saturday Skills for Success program offers workshops and tutoring from 10am-1pm during fall and spring semesters.

Early Alert Services

Early Alert Services (<http://web.uri.edu/earlyalert/>) are part of New Student Programs, and are intended to help students when an intervention of support is needed. In collaboration with other campus support services, we reach out to students to provide individual guidance and help create strategies to address whatever issues may be of concern. Students may need only one meeting with our staff, or may need continuing support. Either way, we are here to help.

Issues that Early Alert Services supports include: Academic Performance, Class Attendance, Lack of Engagement in the Classroom, Connecting with Campus Resources, Family Concerns, Financial Concerns, Lack of Campus Involvement/ Social Isolation, Time Management/Procrastination.

Faculty, staff, peers, and family members play a vital role in student success. As you notice students struggling academically or personally, it is important to refer these students to Early Alert for assistance. Students are welcomed and encouraged to contact the office directly as well. Referrals may be done via referral form, telephone, email, or in person. Referrals to Early Alert may be kept anonymous.