

CS 115 – Introduction to Computer Science 1

Winter 2019 Course Outline

Last revised: January 7, 2019

Course website: www.student.cs.uwaterloo.ca/~cs115/

Course email: cs115@uwaterloo.ca

Course newsgroup/message board: Learn Discussion Boards

Please note that any term-specific content of this document is decided tentatively at the beginning of the term and is subject to change. See the course website for current, up-to-date information

Course Philosophy

CS 115 has been designed with the diverse interests, backgrounds, and goals of students always in mind. As we are aware of the varied ways in which this material might be of use in the future, we have included concepts that are beautiful and universal.

The focus of the course is foundational concepts in computer science (rather than the mechanics of programming) and ways of thinking about computation (rather than solving tasks in a rote fashion). Students will learn skills that are applicable to new languages and environments, not just those used in the course.

Because of the fundamental nature of the concepts taught in CS 115, the course can serve both students whose primary interest in computers is in their use in their chosen areas of study and students intending to minor in CS.

Course Objectives

This course introduces the principles of program design and the fundamentals of computation through functional evaluation.

Intended Audience

CS 115 is intended for students who are familiar with the use of a computer (Web browsing, etc.) but who have no experience with programming.

Related Courses

Prerequisites: None.

Antirequisites: *BME 121, CS 135, 137, 138, 145, CHE 121, CIVE 121, ECE 150, GENE 121, NE 111, MSCI 121, PHYS 236, SYDE 121*

Successor: CS 116.

Hardware and Software

Used in course: Programs are written in subsets of the language Scheme. Student labs are equipped with the DrRacket integrated development environment running on networked personal computers -
- Macintosh computers running OS X. DrRacket is also available on the Unix Math and CS undergraduate course environments. Versions for Windows, Mac OS, Unix/X and Linux are freely downloadable for use on computers owned by students.

References

Textbook (Optional)

The textbook for the course is "How to Design Programs" by Felleisen, Flatt, Fiedler, and Krishnamurthi, MIT Press, 2003. It is available in the UW Bookstore. The entire text is also available on the web at <http://www.htdp.org>. We encourage students to buy the paper version, as the course is built around the textbook and students must read most of the book carefully. If you choose to use only the Web version, please respect the copyright notice; do not print it out. You will not be permitted to access the solutions, so please don't try.

Course notes are also required.

Schedule

Three hours of lecture per week, plus a 80 minute lab.

Communication

Discussion Boards on Learn will be used to communicate with students throughout the term.

General Outline of Topics to be Covered

Module 1 (Introduction) [4.5 hours]: Course information and goals. Functions and evaluation. DrRacket. Defining functions. Syntax and semantics. Substitution.

Module 2 (The Design Recipe) [3 hours]: A design recipe for a function, with a contract, purpose, examples, definition, and tests. Using constants. Helper functions. Strings.

Module 3 (New Types of Data) [3 hours]: Boolean functions. Predicates. Testing using Booleans. Conditional expressions. Designing conditional functions. Templates. Mixed data.

Module 4 (Compound Data: Structures) [3 hours]: Compound data and user-defined structures. Data definition and analysis. Functions that operate on compound data. Templates and data-directed design. The design recipe for compound data. Type predicates.

Module 5 (Lists) [4.5 hours]: Constructing lists. Visualizations of lists. The recursive definition of a list. Functions that process lists. The design recipe for self-referential data definitions. Hazards of not using templates. Nonempty lists. Strings as lists of characters. Lists of structures.

Module 6 (Working with Recursion) [4.5 hours]: A recursive definition of a natural number. Processing natural numbers. Templates for counting down. Templates for counting up. Auxiliary recursive functions. Sorting. Insertion. List abbreviations. Lists containing lists. Association lists. Lists versus structures.

Module 7 (Working with Two Lists or Numbers) [3 hours]: Processing two lists or two numbers. A list going along for the ride. Lock-step. Processing at different rates. Symbols. Symbols versus strings.

Module 8 (Local and Functional Abstraction) [6 hours]: Organizing definitions using local. Encapsulating auxiliary functions and definitions. Lexical scope and block structure. Functional and data abstraction. Lambda.

Module 9 (Binary Trees) [3 hours]: Binary arithmetic expressions. Binary search trees.

Module 10 (General Trees) [1.5 hours]: General arithmetic expressions. Mutual recursion. Leaf-labeled trees.

Course Notes (Lecture Handouts)

The entire set is available in printed and bound form from Media.doc Math (MC 2018); students should purchase a copy and bring it to class.

The handouts contain the text and images of the presentations as prepared in advance. The handouts do not contain everything that the instructor will say, or write on the blackboard, or demonstrate on a computer during the course of the lecture. Instructors may also add their own material. The purpose of the handouts is to relieve students from having to copy everything down, so that they can take additional notes covering what is not in the handouts.

Reading handouts is not an adequate substitute for attending lecture. If you need to miss a lecture, you should contact a classmate who was present and catch up promptly. Presentations may also change slightly at the last minute, and timings are approximate.

Check the errata page for errors in the handouts.

Guides

These guides are also available in printed form, bound with the course handouts (available from Media.doc Math, MC 2018)

The survival guide contains advice on how to do well in CS 115 and avoid common pitfalls. We strongly encourage you to read it through as soon as possible and to attempt to follow the advice it contains.

The style guide contains information on how to create assignment and lab solutions, and how to submit assignments.

[CS 115 Survival Guide PDF](#)

[CS 115 Style Guide PDF](#)

In addition, we have provided a detailed example, from Module 5, of the style required for a function involving data definitions and templates: `gradesexample.rkt`

Instructor-specific materials

Instructors may make available additional material for their lecture sections, such as clarification of specific lectures or extra examples. These additional materials will be posted on LEARN or on the course webpage, as the instructor's discretion.

Lab Exercises

Materials for the weekly labs, including interface files teachpacks and documentation are provided on the course web site. Public tests are provided in order to give students a final check on their work, as the lab work is not collected or marked. They should only be used after students have completed the design recipe, including creating their own tests.

All programming problems will use DrRacket, an instructional development environment tailored for use with the CS 115 textbook. DrRacket is available for Unix (including Linux), OS X, and Windows (95 and up). Labs use Macintosh computers running OS X. We encourage students to download this environment and use it on their own computers. DrRacket is also available on the Unix Math and CS undergraduate course environments (command "drracket" under X11).

You can check your work by requesting a public test. Please see the style guide (PDF) for information on how to name and submit your work.

The links below provide materials for the weekly labs, including interface files. You are strongly encouraged to start preparation before your lab.

Winter 2019 Schedule and Contact Information

Section 001: TTh 10:00-11:20 PHY235

Section 002: TTh 11:30-12:50 PHY235

Section 003: TTh 11:30-12:50 MC2035

Section 004: TTh 10:00-11:20 MC2035

Week	Lectures	Lab	Assignments/Exam (Assignments are due at 10:00 AM)
1: Jan 7-11	Module 01 – Introduction	Lab 01 (getting started)	
2: Jan 14-18	Module 01 (continued)	Lab 02 (Module 01)	A00 due Wed Jan 16
3: Jan 21-25	Module 02 – The Design Recipe	Lab 03 (Module 02)	A01 due Wed Jan 23
4: Jan 28-Feb 1	Module 03 – New Types of Data	Lab 04 (Module 03)	A02 due Wed Jan 30
5: Feb 4-8	Module 04 – Compound Data: Structures	Lab 05 (Module 04)	A03 due Wed Feb 6
6: Feb 11-15	Module 05 – Lists	Lab 06 (Module 05)	A04 due Wed Feb 13
7: Feb 18-22	Family Day and Reading Week	No Lab	No Assignment Due
8: Feb 25-Mar 1	Module 05 Continued Module 06 – Working with Recursion	Lab 07 (Module 05 & 06)	A05 due Wed Feb 27
9: Mar 4-8	Midterm Module 06 (continued) Module 07 – Processing Two Lists or Numbers	Lab 08 (Module 06 & 07)	Midterm Exam: Monday, Mar 4 No Assignment
10: Mar 11-15	Module 07 (continued) Module 08 – Local and Functional Abstraction	Lab 09 (Module 07 & 08)	A06 due Wed Mar 13
11: Mar 18-22	Module 08 (continued)	Lab 10 (Module 08)	A07 due Wed Mar 20
12: Mar 25-29	Module 08 (continued) Module 09 – Binary Trees	Lab 11 (Module 08 & 09)	A08 due Wed Mar 27
13: Apr 1-5	Module 09 (continued) Module 10 – General Trees	Lab 12 (Module 09 & 10)	A09 due Fri April 5

Course Staff Contact Information

Instructors

Joseph Istead

Office: DC3318

E-mail: jwistead@uwaterloo.ca

Collin Roberts

Office: DC3108

E-mail: cdroberts@uwaterloo.ca

Victoria Sakhnini

Office: DC3113

E-mail: vsakhnini@uwaterloo.ca

Instructional Support Coordinator

Barbara Daly

Office: MC4007

Phone: x 36692

E-mail: bmzister@uwaterloo.ca

Lab Instructors

Bettina Boucher

Nisha Eappen

Mbabi Tema

Office Hours: see website

E-mail: cs115@uwaterloo.ca

Grade Calculation

The grading scheme for Winter 2019 is as follows:

- Participation: 5%
- Assignments: 20%
- Midterm: 30%
- Final Exam: 45%
- Bonus Labs: 3%

Notes:

There is one midterm this term, worth 30%.

- You must **pass the weighted exam average** in order to pass the course. For example, if you get 60% on the midterm (18/30) and 40% on the final (18/45), then you will not pass the course since your weighted exam average is 36/75 (less than 50%). This is independent of your assignment grade. If instead you received 40% on the midterm (12/30) and 60% on the final (27/45), then your weighted exam average is 39/75, a passing grade. In this case, you must get at least 11/25 on the assignments and iClicker to pass the course (since your final grade would be 50/100).
- There will be 9 weekly assignments counting for 20% of the final grade, as shown on the Assignments page. Assignment 0 is not worth any marks, but it must be submitted for all other grades to be counted.
 - All assignments will be weighted equally.
 - Your assignment with the lowest mark will be dropped from your final grade, you must however have successfully completed a minimum of 6 assignments and Assignment 0 must be completed (excluding discipline cases).
- Assignments are created by the instructors and are marked by the graduate student teaching assistants based on specifications drawn up by the instructors. The midterm and final are created by the instructors and marked by the instructors, ISAs, and all graduate teaching assistants.
- The best 75% clicker grades will be counted into the grade.
- In order to receive the 3% bonus for labs, labs must be submitted by the due date listed in MarkUs and you must pass the basic grade calculation above (without the bonus), as well as the exam portion of the course.

Grade Appeals

Assignments: send an email to the course account (cs115@uwaterloo.ca) clearly stating what you feel was mis-marked. If you are not satisfied with the resolution, contact the instructor, within 2 weeks of the assignment grade being posted on MarkUs.

If you have problems with the marking of a midterm exam, you need to fill out a Remark Request on Learn with a written explanation within a short period (to be announced) after the exam is made available. In all cases, you should check the posted model solutions to understand your errors.

Course Work Policies

Assignment Submission

MarkUs is used for assignment submission.

Missed or Late Work

Late Assignments: Late assignments will not be accepted; consequently, you should aim to finish early, to allow for unexpected delays. You may submit a single question as often as you wish; submitting a partially-completed solution is a good idea, just in case you don't get back to it in time. The style guide gives information on how to submit your assignments.

After an assignment due date has passed, you may still submit your work for feedback only (no marks) but you must inform the CS115 ISAs by email so they are aware of your submission and request for feedback.

Missed work due to illness: With appropriate, authorized documentation, assignment work may be excused. If a missed assignment is excused, its weight is distributed over the remaining unexcused assignments. In the interest of understanding the course material for future assignments and exams, students who miss work are encouraged to do it, submit it and request feedback from the ISAs.

Corrected assignments: Corrected assignments will be available on MarkUs, usually the week after the assignment is due. The Post-Mortems page will discuss common errors found on assignments. Complete solutions will be posted in the display cases outside MC 4065. If you believe that errors were made in the marking, please email a ISA and state clearly what you feel was mis-marked. The deadline for this is one week after the marks are released.

Group Work

There is no group work.

Other Important Information

Academic Integrity

In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check the Office of Academic Integrity's website, www.uwaterloo.ca/academicintegrity, for more information.

All members of the UW community are expected to hold to the highest standard of academic integrity in their studies, teaching, and research. This site explains why academic integrity is important and how students can avoid academic misconduct. It also identifies resources available on campus for students and faculty to help achieve academic integrity in — and out — of the classroom.

MOSS (Measure of Software Similarities) is used in this course as a means of comparing students' assignments to ensure academic integrity.

Grievance

A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4, <http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm>. When in doubt please be certain to contact the departments administrative assistant who will provide further assistance.

Discipline

A student is expected to know what constitutes academic integrity, to avoid committing academic offenses, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offense, or who needs help in learning how to avoid offenses (e.g., plagiarism, cheating) or about rules for group work/collaboration should seek guidance from the course professor, academic advisor, or the Undergraduate Associate Dean.

For information on categories of offenses and types of penalties, students should refer to Policy 71, Student Discipline, <http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm>. For typical penalties check Guidelines for the Assessment of Penalties, <http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm>.

Avoiding Academic Offenses

Most students are unaware of the line between acceptable and unacceptable academic behaviour, especially when discussing assignments with classmates and using the work of other students. For information on commonly misunderstood academic offenses and how to avoid them, students should refer to the Faculty of Mathematics Cheating and Student Academic Discipline Policy, http://www.math.uwaterloo.ca/navigation/Current/cheating_policy.shtml

Appeals

A decision made or penalty imposed under Policy 70, Student Petitions and Grievances (other than a petition) or Policy 71, Student Discipline may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72, Student Appeals, <http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm>.

Note for students with disabilities: The Office for Persons with Disabilities (OPD), located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the

curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the OPD at the beginning of each academic term.

See [url{http://www.studentservices.uwaterloo.ca/disabilities}](http://www.studentservices.uwaterloo.ca/disabilities) for more information.

Note for students with disabilities

The Accessibility Services Office (AS), located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the AS at the beginning of each academic term.

Intellectual Property.

Students should be aware that this course contains the intellectual property of their instructor, TA, and/or the University of Waterloo. Intellectual property includes items such as:

- Lecture content, spoken and written (and any audio/video recording thereof);
- Lecture handouts, presentations, and other materials prepared for the course (e.g., PowerPoint slides);
- Questions or solution sets from various types of assessments (e.g., assignments, quizzes, tests, final exams); and
- Work protected by copyright (e.g., any work authored by the instructor or TA or used by the instructor or TA with permission of the copyright owner).

Course materials and the intellectual property contained therein, are used to enhance a student's educational experience. However, sharing this intellectual property without the intellectual property owner's permission is a violation of intellectual property rights. For this reason, it is necessary to ask the instructor, TA and/or the University of Waterloo for permission before uploading and sharing the intellectual property of others online (e.g., to an online repository).

Permission from an instructor, TA or the University is also necessary before sharing the intellectual property of others from completed courses with students taking the same/similar courses in subsequent terms/years. In many cases, instructors might be happy to allow distribution of certain materials. However, doing so without expressed permission is considered a violation of intellectual property rights.

Please alert the instructor if you become aware of intellectual property belonging to others (past or present) circulating, either through the student body or online. The intellectual property rights owner deserves to know (and may have already given their consent).