Syllabus

4 September 2019

Welcome to CS168. In this course, we will use the Haskell language to explore the functional programming paradigm. Functional programming makes the mathematical notion of a *function* a centerpiece of our programs. It also eliminates or reduces "side effects" that make programs difficult to reason about and parallelize. We will explore the basics of Haskell, recursion, list processing, patterns and guards, algebraic data types, type classes, combinators, applicative functors, and monads.

When: Monday, Wednesday 12–1:50 PM

Where: M-411

Credits: 3

Prerequisites: CS117

Contact Info

Instructor: Prof. Christopher League, Ph.D.

Email: `christopher.league@liu.edu`[1] — please **include "CS168"** in the subject. I have several email addresses, but all messages end up in the same place, so please use only one.

Office hours: Monday 3–3:50 PM; Wednesday 9–9:50 AM and 3–3:50 PM; and by appointment using `bookme.liucs.net`[2]

Office phone: probably +1 718 488 1137 (but email is much more reliable)

Office location: either Pratt 122 or Sloane 101 (inside School of Business)

Resources

- `liucs.net/cs168f19/`[3] – notes, schedule assignment handouts
- `gitlab.liu.edu`[4] — discussion forum, assignment submission, feedback
- `cchk.in`[5] — attendance

There is no required textbook, but it's a really good idea to have access to one or more books to supplement and for reference. Here are some suggestions:

- **Learn You a Haskell for Great Good!** by Miran Lipovača: buy or read online[6]
- **Haskell: The Craft of Functional Programming** by Simon Thompson: 3rd edition[7] or 2nd edition;[8] either one is helpful
- **Programming in Haskell** by Graham Hutton: 2nd edition[9] or 1st edition;[10] either one is helpful

[1] `christopher.league@liu.edu?subject=CS168`

[2] `bookme.liucs.net/`

[3] `liucs.net/cs168f19/`

[4] `gitlab.liu.edu/`

[5] `cchk.in/`

[6] `learnyouahaskell.com/`

Requirements

Your grade will be computed based on weekly assignments and exams. There are a total of 1,000 points available, broken down as follows:

- There will be **12 assignments**. The exact requirements and expectations for each will be posted to the course web site, but most involve some amount of programming. The assignments are worth **80 points each,** but I will drop the lowest two, so that only ten assignments count, for a total of **800 points.**
- There will be a midterm and final exam, worth **100 points each** for a total of **200 points.**

On the 1,000-point scale, you can expect the following letter grades:

>= 930:	**A**	>= 770:	**C+**
>= 900:	**A–**	>= 730:	**C**
>= 870:	**B+**	>= 680:	**C–**
>= 830:	**B**	>= 600:	**D**
>= 800:	**B–**	else:	**F**

In the end, I may choose to adjust the scale slightly to compensate for assignments or questions that turned out to be trickier than I intended. Such adjustments would never *lower* your grade from what is designated in the above table; if you achieve 930 points, you are guaranteed an **A**.

Policies

It is important to **complete tasks on time,** so you don't fall behind. If you need to miss an exam, try to notify me in advance so we can make other arrangements. **Late assignments** are accepted up until finals week, but will be penalized as follows.

We define a *lateness factor* f as a real number in the range $\{0\ldots1\}$ that will be multiplied by your earned score to determine a late score. The formula is:

$$f = \min\left(1.0, \frac{18 - \log_2\left(\frac{h}{24}\right)}{20}\right)$$

where the variable h represents the number of hours the submission is late. The table below shows some sample values of the late factor for increasingly late submission times.

[7] a.co/7CGcVt1

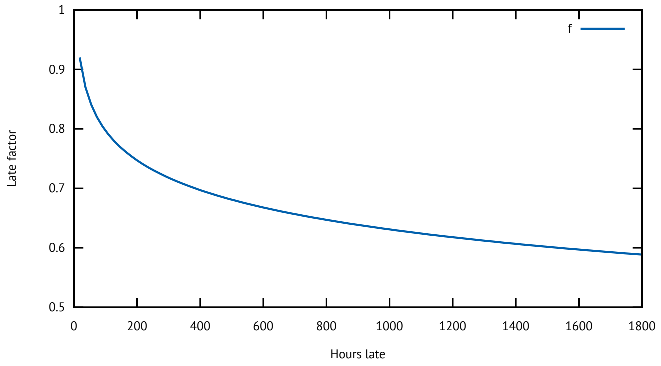
[8] a.co/cmeUeWy

[9] a.co/4WNZ9sZ

[10] a.co/dDf3vQ3

weeks late	days late	hours late (h)	late factor (f)
0.01	0.1	2.4	1.000
0.04	0.3	7.2	0.987
0.07	0.5	12.0	0.950
0.14	1.0	24.0	0.900
0.29	2.0	48.0	0.850
0.43	3.0	72.0	0.821
1.00	7.0	168.0	0.760
2.00	14.0	336.0	0.710
4.00	28.0	672.0	0.660
8.00	56.0	1344.0	0.610

The idea is that is that the penalty is somewhat steep initially (from an **A** to a **B+** after just one day) but shallows out over time. It will still be worthwhile to submit a missing assignment, even weeks late.



There will be no extra credit. Students usually ask for extra credit late in the semester after they have already squandered their original opportunities. Be sure to start your work early, so that we can detect and solve any problems before they can impact your grade.

Plagiarism is the use or presentation of ideas, words, or work that is not one's own and that is not common knowledge, without granting credit to the originator. Plagiarism is a practice that is not only unacceptable, but which is to be condemned in the strongest terms possible on the basis of moral, educational and legal grounds. Under University policy, plagiarism may be punishable by a range of penalties from a failing grade in the assignment or course to dismissal from the School of Business, Public Administration and Information Sciences. All students are required to read the handbook on avoiding plagiarism.[11]

Cheating includes, but is not limited to the following: falsification of statements or data; listing sources that have not been used; having another individual write your paper or do your assignments; writing a paper or creating work for another student to use without proper attribution; purchase of paper or research work for one's submission as their own work; using written, verbal, or electronic or other sources

[11] `liucs.net/u2`

of aid during an examination (except when expressly permitted by the instructor, depending on the nature of the examination) or knowingly providing such assistance to aid other students.

In a course with programming assignments, it is usually okay to work with and learn from other students to **some** extent, but what you submit in the end needs to be your own. The most reliable way to do that would be to set aside whatever code you created together, and then recreate it from scratch on your own.

Showing up on time to class is extremely important. If you must be absent or more than 5 minutes late, please try to notify me in advance. I will be keeping track of whether you are in class, and when you arrive. A few missed classes will not count against you, but habitual absence will significantly hurt your grade. Additionally, there will be no make-up quizzes. I do not distinguish between “excused” and “unexcused” absence. Unless you miss an **exam** due to a severe medical emergency, I don’t need to see a doctor’s note. If you do miss an exam, the make-up exam may be somewhat different from the one given in class.

In accordance with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990, including changes made by the Americans with Disabilities Amendments Act of 2008, the Long Island University **does not discriminate against qualified individuals with disabilities.** If you are a student with a documented disability/impairment (psychological, neurological, chronic medical, learning disability, sensory, physical) and require reasonable accommodations, please register with Student Support Services and provide me with an accommodation letter. Visit Sloan Building 1st floor, call 718 488 1044, or visit Student Support Services.[12]

I participate in the **LIU Safe Zone** program. Representatives of the program serve as contacts for individuals on campus with questions or concerns related to sexual orientation and gender identity, whether of self or of a friend or family member. The goal of the program is to promote a safe and free campus for all students. Safe Zone areas can be identified by a sticker with the LIU Safe Zone logo.

The **Family Educational Rights and Privacy Act** (FERPA) gives students control over the disclosure of their educational records. During this course you may have the opportunity to create accounts or register with certain public online services. In these cases, you need not make any personally identifying information public. You may use a pseudonym or online handle, as long as you identify yourself to the instructor.

[12] `www.liu.edu/Brooklyn/SSS`

Schedule

***Wed 4 Sep*: Meeting 1**
What is FP? Intro to REPL and evaluating expressions.

***Mon 9 Sep*: Meeting 2**
Tuples, Booleans, partial application, operator sections, pattern guards.

***Tue 10 Sep*: Assignment 1**

***Wed 11 Sep*: Meeting 3**
Recursive functions, evaluating expressions.

***Mon 16 Sep*: Meeting 4**
List constructors and pattern matching.

***Tue 17 Sep*: Assignment 2**

***Wed 18 Sep*: Meeting 5**
Implementing list operators, syntax of list comprehensions.

***Mon 23 Sep*: Meeting 6**
Intro to type syntax and some built-in type classes.

***Tue 24 Sep*: Assignment 3**

***Wed 25 Sep*: Meeting 7**
Type classes including `Show`, `Eq`, `Ord`, `Enum`, `Bounded`. Defining type synonyms and enumerated types. Brief intro to `Maybe` type.

***Mon 30 Sep*: Meeting 8**
More on the `Maybe` type, including `lookup` and `fmap`. The `case` expression, field selectors, and recursive data types.

***Tue 1 Oct*: Assignment 4**

***Wed 2 Oct*: Meeting 9**
Function composition using `.` and `<=<`, the `Either` type.

***Mon 7 Oct*: Meeting 10**
Defining a bounded stack type, `Functor` and `Bifunctor` type classes.

***Tue 8 Oct*: Assignment 5**

***Wed 9 Oct*: Meeting 11**
The `Monoid` type class.

***Mon 14 Oct*: Meeting 12**
The `Ordering` type, laziness, infinite data structures.

***Tue 15 Oct*: Assignment 6**

***Wed 16 Oct*: Meeting 13**
Data types for abstract syntax trees.

***Mon 21 Oct*: Meeting 14**
Variable environments for expression trees, applicative functors.

***Wed 23 Oct*: Meeting 15 [exam]**
Midterm exam.

***Mon 28 Oct*: Meeting 16**
Pseudo-random number generation and threading state.

***Tue 29 Oct*: Assignment 7**

***Wed 30 Oct*: Meeting 17**
Generalizing PRNG operations, leading to `always` / `andThen`.

***Mon 4 Nov*: Meeting 18**
Monad operations for `Maybe` and List.

***Tue 5 Nov*: Assignment 8**

***Wed 6 Nov*: Meeting 19**
Remaining topics TBD…

***Mon 11 Nov*: Meeting 20**

***Tue 12 Nov*: Assignment 9**

***Wed 13 Nov*: Meeting 21**

***Mon 18 Nov*: Meeting 22**

***Wed 20 Nov*: Meeting 23**

***Mon 25 Nov*: Meeting 24**

***Tue 26 Nov*: Assignment 10**

***Mon 2 Dec*: Meeting 25**

***Wed 4 Dec*: Meeting 26**

***Mon 9 Dec*: Meeting 27**

***Tue 10 Dec*: Assignment 11**

***Wed 11 Dec*: Meeting 28**

***Mon 16 Dec*: Assignment 12**