

CS 168 Syllabus

5 September 2018

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 12:00–1:50pm

Where: M412

Credits: 3

Contact Info

Instructor: Prof. Christopher League, Ph.D.

Email: christopher.league@liu.edu — please include the course number (CS168) in the subject. I have several email addresses, but all messages end up in the same place, so **please use only one**.

Instant Messaging: cleague@gmail.com (Google Hangout)

Office hours: Monday, Wednesday 3–3:50 PM, or make an appointment at <https://bookme.liucs.net/>

Office phone: +1 718 488 1274 (voice mail only)

Office location: H-700, LIU Brooklyn

Resources

We will use several web resources:

- <https://liucs.net/cs695f17/> — notes, schedule
- <https://class.mimir.io/> — assignment platform
- <http://www.gradechamp.com/> — grade reports

There is an required extra fee for the Mimir platform, which we will use for assignments. You can purchase access at that link with a credit card, or buy a pass from the campus book store. Once you join, the course access code is 596d9b87b2.

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

- **Learn You a Haskell for Great Good!** by Miran Lipovača: <http://learnyouahaskell.com/> (buy or read online for free)
- **Haskell: The Craft of Functional Programming** by Simon Thompson: <http://a.co/7CGcVt1> (3rd edition) or <http://a.co/cmeUeWy> (2nd edition; either one is helpful)
- **Programming in Haskell** by Graham Hutton: <http://a.co/4WNZ9sZ> (2nd edition) or <http://a.co/dDf3vQ3> (1st; either one is helpful)

Requirements

There are a total of 1,000 points available, broken down as follows:

- There will be **12 weekly 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:

	≥ 870: B+	≥ 770: C+		
≥ 930: A	≥ 830: B	≥ 730: C	≥ 600: D	
≥ 900: A–	≥ 800: B–	≥ 680: C–	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 milestone requirements **on time**, so you don't fall behind. Late work will be graded as follows.

This formula specifies a *lateness factor* f that is multiplied by your earned score to determine a late score. The variable h represents the number of hours the submission is late.

$$f = \frac{8.5 - \log_2\left(\frac{h}{24}\right)}{10}$$

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 affect 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 by visiting <https://liucs.net/u2>

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 his/her own work; using written, verbal, or electronic or other sources 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 want to see a doctor's note. If you do miss an exam, the make-up exam will be different – and probably *not* easier.

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 <http://www.liu.edu/Brooklyn/SSS>

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.

Schedule

The day-by-day schedule is shown below, including all deadlines.

- Wed Sep 5 Meeting 1** at 12 pm. What is FP? Intro to REPL and evaluating expressions.
- Mon Sep 10 Meeting 2** at 12 pm. Tuples, Booleans, partial application, operator sections, pattern guards.
- Wed Sep 12 Meeting 3** at 12 pm. Recursive functions, evaluating expressions.
- Sun Sep 16 Assignment 1** due at 23:59.
- Mon Sep 17 Meeting 4** at 12 pm. List constructors and pattern matching.
- Wed Sep 19 Meeting 5** at 12 pm. Implementing list operators, syntax of list comprehensions.
- Sun Sep 23 Assignment 2** due at 23:59.
- Mon Sep 24 Meeting 6** at 12 pm. Review solutions for list assignment, intro to type syntax and some built-in type classes.
- Wed Sep 26 Meeting 7** at 12 pm. Type classes including Show, Eq, Ord, Enum, Bounded. Defining type synonyms and enumerated types. Brief intro to Maybe type.
- Mon Oct 1 Meeting 8** at 12 pm. More on the Maybe type, including lookup and fmap. The case expression, field selectors, and recursive data types.
- Tue Oct 2 Assignment 3** due at 23:59.
- Wed Oct 3 Meeting 9** at 12 pm. Function composition using `.` and `<=<`, the Either type.
- Mon Oct 8 Meeting 10** at 12 pm. Defining a bounded stack type, Functor and Bifunctor type classes.
- Tue Oct 9 Assignment 4** due at 23:59.
- Wed Oct 10 Meeting 11** at 12 pm. Solutions to assignment 4, Monoid type class.
- Mon Oct 15 Meeting 12** at 12 pm. The Ordering type, laziness, infinite data structures.
- Wed Oct 17 Meeting 13** at 12 pm. Help on assignment 5, abstract syntax tree.
- Fri Oct 19 Assignment 5** due at 23:59.
- Sun Oct 21 Assignment 6 SKIPPED** due at 23:59.
- Mon Oct 22 Meeting 14** at 12 pm. Variable environments for expression trees, applicative functors.

Wed Oct 24 Meeting 15 at 12 pm. Stack evaluation, compiling expressions to stack instructions.

Mon Oct 29 Meeting 16 at 12 pm. Midterm exam.

Tue Oct 30 Assignment 7 due at 23:59.

Wed Oct 31 Meeting 17 at 12 pm. Solutions to assignment 7. PRNG and threading state.

Mon Nov 5 Meeting 18 at 12 pm. Generalizing generator operations, leading to `always / andThen`.

Tue Nov 6 Assignment 8 due at 23:59.

Wed Nov 7 Meeting 19 at 12 pm. Solutions to assignment 8. Monad operations for `Maybe` and `List`.

Sun Nov 11 Assignment 9 due at 23:59.

Mon Nov 12 Meeting 20 at 12 pm.

Wed Nov 14 Meeting 21 at 12 pm.

Mon Nov 19 Meeting 22 at 12 pm.

Mon Nov 26 Meeting 23 at 12 pm.

Wed Nov 28 Meeting 24 at 12 pm.

Sun Dec 2 Assignment 10 due at 23:59.

Mon Dec 3 Meeting 25 at 12 pm.

Wed Dec 5 Meeting 26 at 12 pm.

Mon Dec 10 Meeting 27 at 12 pm.

Tue Dec 11 Assignment 11 due at 23:59.

Wed Dec 12 Meeting 28 at 12 pm.

Sun Dec 16 Assignment 12 due at 23:59.