

Syllabus

General Info

Course CS320 Functional Programming

Instructor Charilaos Skiadas (skiadas at hanover dot edu)

Term Winter 2019-2020

Office SCH 111 / LYN 108

Office Hours LYN108: MWF 12:00pm-1pm, T 10:00am-12:00pm and by appointment.

Book *Haskell: The craft of functional programming* by Simon Thompson, 3rd edition. ISBN: 978-0-201-88295-7

Websites • book website: <http://www.haskellcraft.com/>¹
• notes².

Class times MWF 10:40am-11:50am in LYN120A.

Course Description

Functional Programming (FP) is a particular approach to programming that has gained increased popularity in recent years, even though its origins date back to the early days of programming.

In FP functions take center stage. We move programs along by feeding input into functions and processing their output, often by feeding it into yet other functions. While this is to a large extent true in other programming styles, it becomes a dominant feature in FP. As an example, in most languages a sorting *procedure* will be given an array as input and rearrange its entries in place to produce a sorted array. By contrast, a sorting *function* in an FP language would instead return a new array with the values sorted, and keep the initial array intact.

In this course we will learn about the FP programming style, and its strengths and weaknesses. We will do so in the context of the Haskell programming language, which apart from being a quintessential FP language offers many other interesting features like *type classes* and *lazy evaluation* and a complete lack of *mutation*.

¹<http://www.haskellcraft.com/>

²skiadas.github.io/FunctionalProgrammingCourse/

Course objectives

In this course you will learn the essential characteristics of functional/effect-free programming, namely:

- using functions without side-effects and employing compositional reasoning to build new functions by chaining together other functions.
- working with immutable variables and understanding the benefits that this provides as well as the challenges it presents.
- using pattern-matching to process structured data.
- using functions as first-class values, as useful inputs and outputs of other functions.
- understanding the effects of function closures and the programming idioms they enable.
- working with higher-order operations on aggregate structures.

You will also delve more into a number of concepts relating to type systems:

- the use of basic and compound types in Haskell, including function types.
- goals and limitations of static typing.
- type inference.
- generic types and parametric polymorphism vs ad-hoc polymorphism.

Course Components

Reading Notes

On the website you will find a schedule³ with links to documents for each class day. In those documents you will find notes for the day's lesson, and reading assignments.

Class Attendance

You are expected to attend every class meeting. You are only allowed to miss 3 classes without excuse. From that point on, every unexcused absence will result in a reduction of your final score by one percentage point, up to a total of 5 points. Excused absences should be arranged in advance, and backed by appropriate documentation. Emergencies will be dealt with on an individual basis. There are very few reasons that would qualify as an excuse for an absence.

³<http://skiadas.github.io/FunctionalProgrammingCourse/schedule.html>

Programming Assignments

There will be regular programming assignments. The assignments are there to help you practice your understanding of new concepts, as well as to prepare you for the major project. Programming assignments are 35% of your final grade.

Exams

There will be one midterm, tentatively scheduled for Friday, February 14th and a final during finals week. **You have to be here for the exams.** If you have conflicts with these days, let me know as soon as possible. Do not plan your vacation before you are aware of the finals schedule.

Programming Project

There will also be a larger programming project, that we will discuss in more detail later in the term. It will require that you work together with your classmates, especially towards the end of the semester. It will count for 30% of your final grade.

Getting Help

- You should never hesitate to ask me questions. I will never think any less of anyone for asking a question. Stop by my office hours or just email me your question, which has the great benefit of forcing you to write it down in clear terms, which often helps you understand it better.
- You are allowed, and in fact encouraged, to work together and help each other regarding the notes and the theory. You can also discuss *general* questions about the programming assignments. But I expect you to work on the programming assignments on your own.

Grading

Your final grade depends on class attendance, homework, project, quizzes, midterms and the final, as follows:

Component	Percent
Attendance	5%
Assignments	35%
Project	30%
Midterm	15%
Final	15%

This gives a number up to 100, which is then converted to a letter grade based roughly on the following correspondence:

Letter grade	Percentage Range
A, A-	90%-100%
B+, B, B-	80%-90%
C+, C, C-	70%-80%
D+, D, D-	60%-70%
F	0%-60%