

CSCE 314: Programming Languages

Dr. Dylan Shell

Class Overview

CSCE 314: Programming Languages

- **Course Homepage:**

<http://robotics.cs.tamu.edu/dshell/cs314/>

- **Textbooks:**

1. Programming in Haskell, Graham Hutton, Cambridge University Press (Second Edition)
2. The Java Programming Language, 4th Ed, Ken Arnold, James Gosling, and David Holmes, Addison- Wesley Professional

<http://www.informit.com/store/java-programming-language-9780321349804>

- **Other reading material: See course homepage**

CSCE 314: Programming Languages

- **CSCE 221 (?)**
- **CSCE 222 (not official, but desirable)**
- **Responsibility**
- **Familiarity with recursion**
- **Familiarity with complexity analysis**
- **Eager to learn (Energy and enthusiasm)**
- **Perseverance and positive, constructive thinking**
- **...**

Motivation for the course

- **Ever increasing complexity and the size of modern computer programs**
- **Ever increasing time and cost of program development**
- **Ever increasing requirements on the correctness of the programs**
- **Ever increasing need for a better programmer who learns new languages easily!**

**“We shape our tools, and
then our tools shape us.”**

Marshall McLuhan (or John Culkin?)

Course Objectives (1)

Understand the fundamental concepts of the programming languages:

- Broader understanding of language constructs, common abstraction mechanisms, and efficiency consideration
- Understanding of the basics of how programs written in high-level programming languages are executed, i.e., parsing, internal program representation, type checking, interpretation

Course Objectives (2)

Understand the fundamental concepts of the programming languages:

- Effective use of the functional programming approach to design and implement programs
- Understanding of the main features of modern object-oriented languages
- (Towards) Encouraging the use of formal verification

Further introductions

Flemming Andersen, Ph.D.

Formerly Principal Engineer and formal verification manager at Intel. Managed the formal verification (FV) of the RTL in the Xeon-Phi processors. The main focus is on arithmetic verification since we never want to encounter a new FDIV bug like the one that cost Intel almost \$500 million in 1994.

Mengyuan Chao (TA)

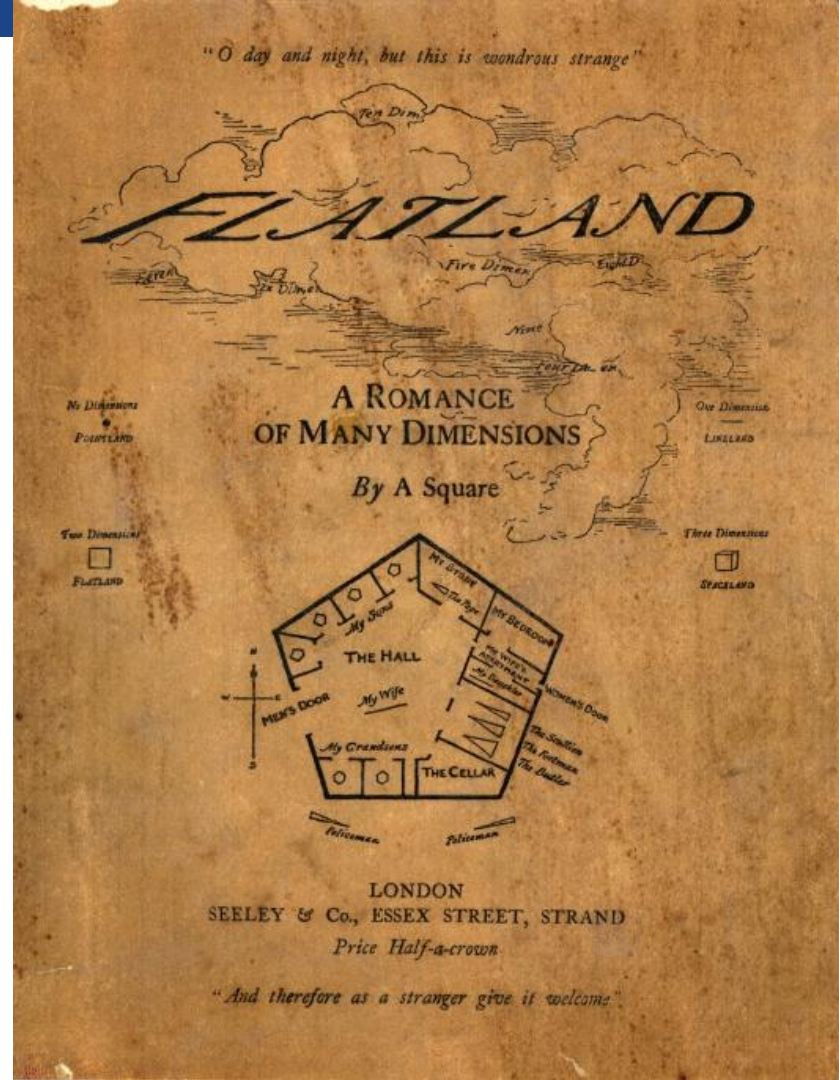
Has a great deal of a experience in working with Haskell; his dissertation research deals with networking, he has published papers on QoS for IP networks.

Edwin A. Abbott's Flatland



Read it, its free!

<https://ebooks.adelaide.edu.au/a/abbott/edwin/flatland/>



Course Outline (1)

Language Processing

- Grammars, lexing and parsing
- Abstract syntax, internal representations
- Types and type checking
- Interpretation
- Basics of an implementation of a simple programming language

Course Outline (2)

Study of a Functional Language — Haskell

- Type inference
- Parametric polymorphism
- Higher-order functions
- Algebraic data types
- Abstract data types and modules
- Type classes
- Effects in a “pure” language

Course Outline (3)

Study of a Functional Language — Java

- Subtyping and inheritance, subtype polymorphism
- Exception handling
- Generics, wildcards
- Reflection
- Concurrency

Grading

Midterm Examination: 40% (In class)

Final Examination: 60% (Reserved final slot)

- The final examination will be comprehensive

Examinations will be closely related to problems posted as homework.

For both exams students are permitted 5 pages of notes.

* The honors section for this course will have an additional assignment that will account for 15% of their grade; the two exams will be scaled proportionately.

Homeworks

- Homework will be posted approximately fortnightly.
- Best attempt the assignments individually; if you need help, ask your friends. If approached, help!
- Problems may involve material that is being covered or is to be covered, but the majority will use material already discussed.
- Much of the in-class discussion will build on material in the homework.

Collaboration Policy

- Best attempt the assignments individually; if you need help, ask your friends. If approached, help others!
- Use piazza: post, reply, discuss...
- Help is available from peer teachers.
- Manufacture circumstances using social pressures to get homework completed.

Earning grade in this course

- **Planning to get an “A” grade?**

Attend all classes, pay attention and participate in the class, **do the homework exercises**, help others, read book and reading material, participate in class.

- **How to get a “D” grade?**

Miss classes, do not pay attention in the class, do not do assignments, don't read the reading materials.

Summary

- **We will study fundamentals of programming languages by way of learning two languages — Haskell and Java**
- **A lot of “fun” work in the class**

Credits and Image sources:

- <http://www.mathaware.org/mam/00/master/people/abbott/JPG/abbott.jpg>
- <https://ebooks.adelaide.edu.au/a/abbott/edwin/flatland/>

These slides and most that will be presented are based on materials kindly provided by Dr. Hyunyoung Lee and Dr. Jaakko Järvi. I acknowledge their help and graciousness.